2.28 RANGE TRACKING

The range track functional element for monopulse systems has two subelements: (1) the receiver functions that measure the range track error, and (2) the range-gate positioning servo (in the case of analog systems) or range track filters (in the case of digital systems) which act to correct the range errors and thus maintain target tracking. This document is confined to the three-channel and two-channel monopulse radars modeled by ESAMS 2.7; the older track-while-scan (TWS) systems are described in another document.

The split-gate methodology used for range-tracking in fire control radars is also applicable to some missile seekers and TWS radars, and the occurrence of receiver routines which are common with the ground monopulse radars and the seekers and TWS systems will be noted in this document. However, no detailed discussion for seekers and TWS systems will be given here because they are treated elsewhere (e.g., the seeker analogue to the range-gate servo or track filter subelements are the missile guidance components that are in the missile flyout functional elements.)

In the split range-gate methodology (see [10] for generalities and [37] for ESAMS specifics), the range gate is notionally divided into two halves, called the early and late gates (corresponding to shorter and longer ranges). The energy in the signals in the early and late gates is integrated separately for each half gate, the signals are weighted by the respective gate weights, and the difference is used to determine the error signal. This error signal is converted to meters and normalized to the total range-gate signal. The result is used to drive the gate positioning servo or track filter.

All ground tracking radars in ESAMS 2.7 now use the waveform-driven model of waveform generation, except for the TWS systems. For all non-TWS tracker radars (i.e., all monopulse ground trackers), the specific threat system has a characteristic range-servo. ESAMS 2.7 provides many different models for track servo or track filter. This CMS will illustrate by focusing on two example cases. This section discusses the overall framework for range track, covering the general three- and two-channel monopulse receiver functions dealing with the measurement of range track errors, and discussing in detail the range track filter model most used in ESAMS 2.7. Because range track filters are characteristic of digital radars, such as more modern systems, this section exemplifies digital range track. A classified addendum (to be published at a later date) will deal with a special detailed two-channel monopulse receiver and the special range-gate-positioning servo associated with it. These models exemplify analog range track. The classified addendum will also treat some range track electronic counter-countermeasures (ECCM).

2.28.1 Functional Element Design Requirements

This section contains the design requirements necessary to fully implement the monopulse range track simulation.

a. ESAMS will calculate the range error signal produced by the receiver from the output of the monopulse antenna system.

b. ESAMS will simulate the detailed receiver functions related to range track for the detailed two-channel monopulse receiver.

c. ESAMS will simulate the detection of electronic countermeasures (ECM) affecting range-track and simulate the application of appropriate electronic counter-countermeasures (ECCM).

d. ESAMS will simulate the effect of producing updated/predicted range coordinate and range-rate from the range error output from the receiver, either through range-gate positioning servos or through range track filters.

Requirements b and c will be treated in detail in the classified addendum, as is the analog range servo.

These requirements must be implemented in sufficient fidelity to reflect the effects of extraneous influences, such as noise, clutter and multipath, ECM, and target fluctuations, as well as the high range rates that can be encountered in tactical scenarios. The range track function is very important in determining whether or not the target aircraft is intercepted. The placement of the range-gate determines which signals are passed on to impact Doppler and angle track, with the latter being especially crucial to missile system performance.

2.28.2 Functional Element Design Approach

This section describes the design approach (equations, algorithms, and methodology) implementing the design requirements of the previous section.

Figure 2.28-1 illustrates a range track loop. It shows the two major subelements embedded in the loop: The range discriminator which measures the range error, and the track range correcting subelement which is either a servo (that repositions the range gate) or a track filter (that updates the range value for the track).

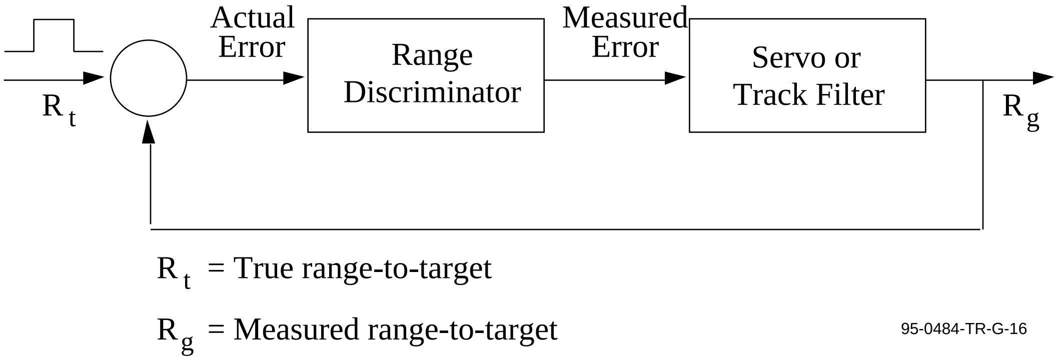

FIGURE 2.28-1. Range Track Loop.

Design Element 28-1: Calculation of Range Error Signal by the Receiver

The range discriminator is the subelement that determines the error between true target range and the current range measurement represented by the range-gate center, and sends this measured range error to the track range correcting subelement. A split-gate discriminator is illustrated in Figure 2.28-2. In this representation, the imbalance in energy between the early gate and the late gate of the split-gate tracker is used to generate the error signal. The error signal will drive the servo or track filter, and the track range will be updated to align with the perceived target range. (Reference [14], pages 351-353, describe the procedure of servo repositioning of the range-gate, and [38], pages 21-25, describe the procedure of fixed coefficient track filter updating.)

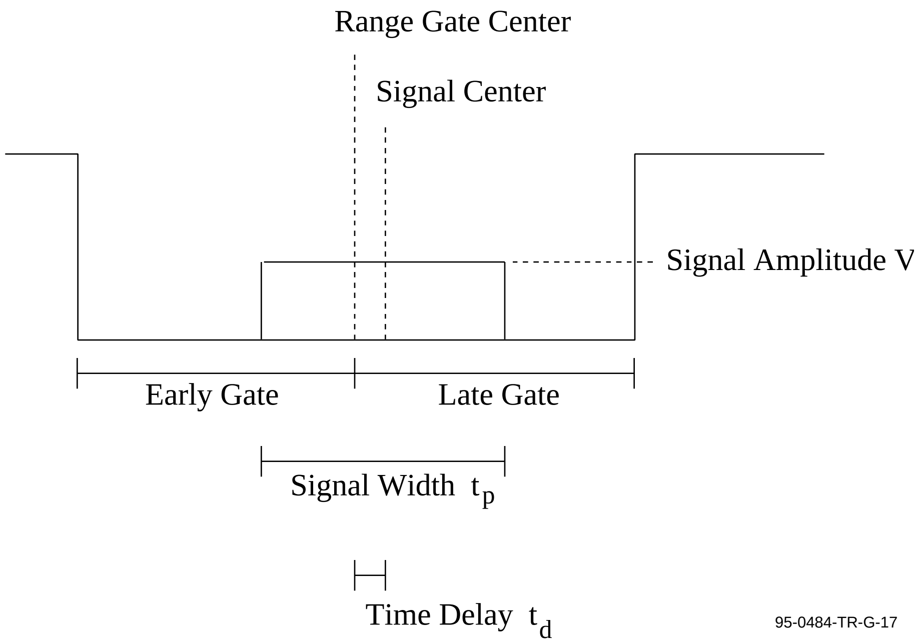

FIGURE 2.28-2. Range Discriminator.

For the simple signal representation shown in Figure 2.28-2, we can develop the range error as follows. Let S_E be the magnitude of the integral of the signal in the early gate; for Figure 2.28-2, it would be V×(t_p/2 - t_d), where V is the magnitude of the voltage. Let S_L be the magnitude of the integral of the signal in the late gate; for Figure 2.28-2, it would be V×(t_p/2 + t_d). Then in terms of the signal integrals in the early and late gates, we can recover the time error t_d as

$$t_d = - t_p \times (S_E - S_L)/(S_E + S_L). \qquad [2.28\text{-}1]$$

However, in general we cannot know the signal width t_p; rather, the best we know is the range gate width and its half-width, t_{gw}/2. Hence, equation (2.28-1) is replaced with equation (2.28-2):

$$t_d = - 0.5 t_{gw} \times (S_E - S_L)/(S_E + S_L). \qquad [2.28\text{-}2]$$

Note that this form is conservative in that if all the signal is in one side of the split gate, say the late gate, equation (3.1-2) says that the error $t_d = 0.5t_{gw}$, which is the extreme late edge of the range gate. ESAMS 2.7 provides for the possibility of weighting the early and late gates differently, say with a weight w_E for the early gate and w_L for the late gate. Then equation (2.28-2) is extended to:

$$t_d = - 0.5 t_{gw} \times (w_E S_E - w_L S_L)/(w_E S_E + w_L S_L). \qquad [2.28\text{-}3]$$

The range error in distance units, ΔR, is obtained from t_d by the standard time to range conversion factor of one-half the speed of light, c (the one-half accounting for the two-way propagation on the time base):

$$\Delta R = - 0.5\times(0.5t_{gw}/c)\times(w_E S_E - w_L S_L)/(w_E S_E + w_L S_L). \qquad [2.28\text{-}4]$$

The quantity $(0.5t_{gw}/c)$ is called RKDIFF in ESAMS.

The discussion so far has dealt only with a single rectangular signal pulse. All signals involved in ESAMS are treated as rectangular pulses with characteristic voltage amplitude, center along the time base, width along the time base, and a phase. The total signal in the range gate will therefore be a superposition of all such signals, from such sources as direct target skin return, multipath return, clutter, thermal noise, ECM sources, and a blanking pulse in the receiver. (Strictly speaking, continuously distributed sources like thermal noise will simply take on the width of the range gate.) In general, then, the simulation must deal with a variety of signals of different locations along the time base, different widths, different magnitudes, and different phases. Figure 2.28-3 illustrates this variety for a number of pulses having different relationships to the edges of the range gate.

The integration of the signals in the range gate proceeds in two steps: (1) First the signals on the signal bus (for the sum channel) are considered, and the range gate is partitioned into intervals marked by the locations of all pulse edges occurring in the range gate. Then (2) the integrated signal in each of the split gates is obtained by summing all complex signals in each interval for each split gate, with the contribution from each interval weighted by the width of the interval; this results in the quantities S_E and S_L of equation (2.28-3). In the integration, any contributions that lie within the blanking pulse edges (if the blanking pulse is on and any part is in the gate) are skipped. Finally, the split gate weights are applied and equation (2.28-2) evaluated to get the range error, RERROR, which is passed to the range-correcting subelement.

Figure 2.28-3 can be used to illustrate the way the integration proceeds. The partitioning of the range gate for the signals in this figure identifies six edges identified in parentheses in the figure. The early (or left) edge of the range gate (TRGL) is always edge number 1. Pulses 1 and 2 are completely outside the range gate, so they contribute no edges. Pulse 3 is partially in the gate; its right (late) edge is partition edge number 3. Pulse 4 is entirely in the gate, with its left edge to the left of the end of pulse 3; this left edge of pulse 4 is partition edge 2. The right edge of pulse 4 is partition edge 4. Pulse 5 is partially in the gate with its left edge to the right of all other edges except the right edge of the gate; it is partition edge 5. And the right edge of the range gate is partition edge 6 (the value for the total number of partition edges, NEP). Pulse 6 completely straddles the gate; it is considered to be trimmed to fit the gate, so its left edge coincides with the partition edge 1 and its right edge coincides with partition edge 6. The integration intervals are indexed by the partition edge number on the left of the interval. Let us assume that none of the pulses in the figure are blanking pulses; then in the integration stage, interval 1 (between edges 1 and 2) contributes to the early gate integral the complex sum of the voltages for pulses 3 and 6 times the width of the interval; interval 2 contributes the complex sum of voltages for pulses 3, 4, and 6 times the width of the interval. The early gate integral concludes with a contribution from the complex sum of voltages from pulses 4 and 6 times the distance from edge 3 to the range-gate center (TRGC); the magnitude of this complex sum of sums is S_E. The integral for the late gate begins with a contribution from the complex sum of voltages for pulses 4 and 6 times the distance from the range-gate center to edge 4. Next, interval 4 contributes the complex voltage for pulse 6 times the width of the interval. The late gate integral concludes with a contribution from the complex sum of voltages from pulses 5 and 6 times the distance from edge 5 to the range-gate right edge (TRGR); the magnitude of this complex sum of sums is S_L.

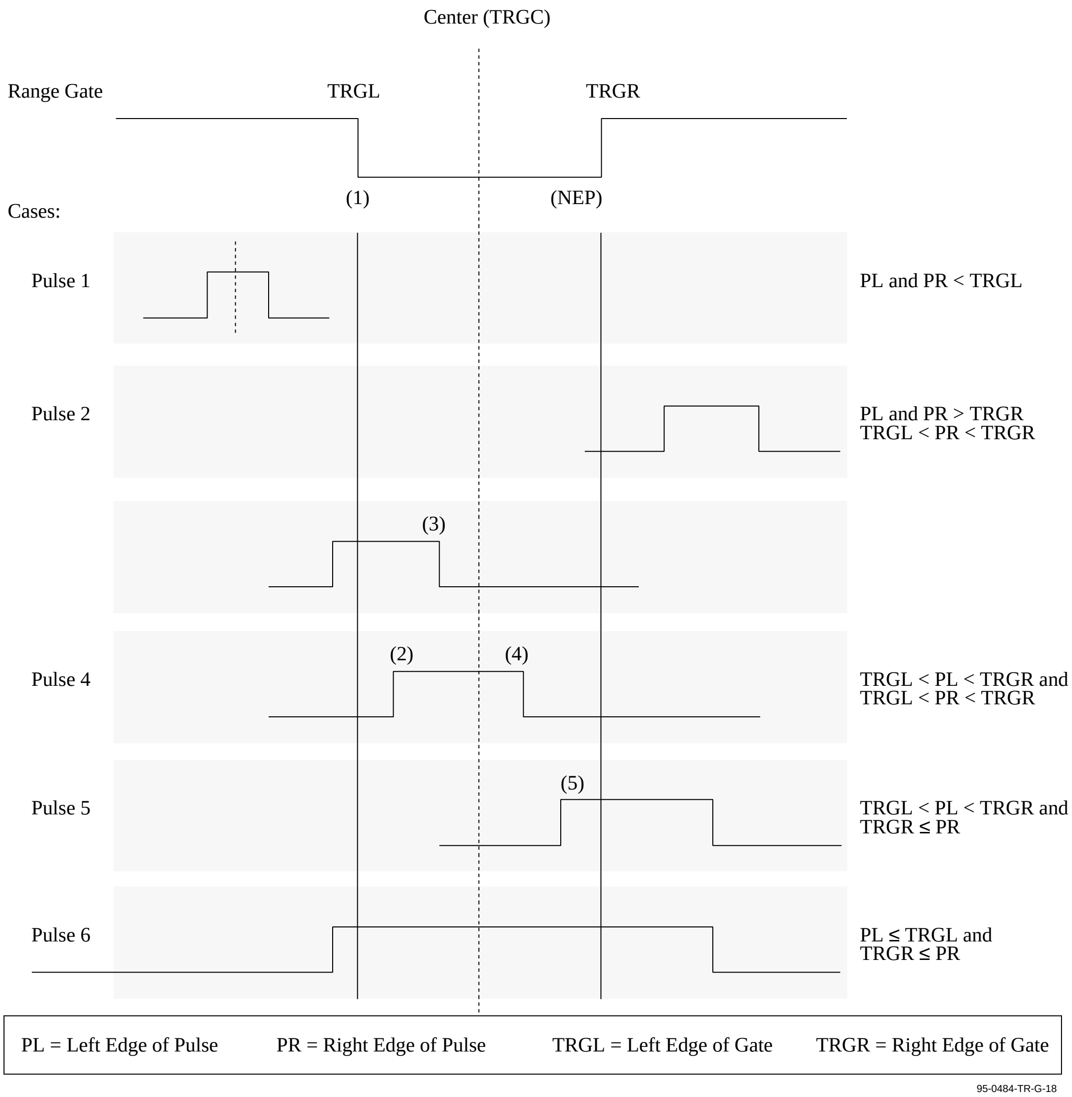

FIGURE 2.28-3. Pulse Positions with Respect to the Range Gate Edges.

Design Element 28-2: Calculate the g, h and k Filter Gains

The filter gains in the g-h-k filters of ESAMS 2.7 are fixed; [38] and articles cited therein discuss various interrelationships between the gains for certain conditions. In ESAMS 2.7, the coordinate gain g is input in the RDRD group, and the velocity and acceleration gains are computed from them according to the following relationships which are widely used (see [13]):

$$g = \text{RDRD input} \qquad [2.28\text{-}5]$$

$$h = \{2g^3 - 4g^2 + ([4g^6 - 64g^5 + 64g^4]\}/[8(1 - g)] \qquad [2.28\text{-}6]$$

$$k = [h(2 - g) - g^2]/g \qquad [2.28\text{-}7]$$

Design Element 28-3: Calculate New Range Gate Position and Range Rate for a g,h,k Track Filter

The fundamental equations of the g-h-k filter are the following ([38], p. 22):

For the smoothed coordinate, velocity, and acceleration:

$$x_s(n) = x_p(n) + g[x_m(n) - x_p(n)] \qquad [2.28\text{-}8]$$

$$v_s(n) = v_s(n\text{-}1) + T\,a_s(n\text{-}1) + h(1/T)[x_m(n) - x_p(n)] \qquad [2.28\text{-}9]$$

$$a_s(n) = a_s(n\text{-}1) + k(1/T^2)[x_m(n) - x_p(n)] \qquad [2.28\text{-}10]$$

and for the predicted coordinate:

$$x_p(n+1) = x_s(n) + T\,v_s(n) + (T^2/2)\,a_s(n) \qquad [2.28\text{-}11]$$

where

$x_s(n)$, $v_s(n)$, and $a_s(n)$ are the smoothed quantities for the n-th sample,
$x_p(n)$ is the coordinate predicted for the n-th sample,
$x_m(n)$ is the measured coordinate at the n-th sample,
T is the time interval between samples, and
g, h, and k are the filter gains for the coordinate, velocity, and acceleration.

For monopulse trackers, the quantity $x_m(n)$, hence $[x_m(n) - x_p(n)]$, is determined from the range error output from the receiver. The range coordinate predicted one time interval into the future, $x_p(n+1)$, is the analogue of the corrected range output from the range-gate servo for analog systems. In ESAMS 2.7, the sample time T is the waveform group-duration time GRPDUR.

The general g-h-k filtering in ESAMS 2.7 provides for two possible values (specified in RDRD input) to be used for the gain g, according to whether the time in the tracking phase is less than or greater than a gain switchover time specified in the RDRD input. For range, the RDRD switchover time is RANTSW, and the early and late gain values are RANGHK(1) and RANGHK(2), respectively.

The new range gate position is $x_p(n+1)$, and the range rate is $v_s(n)$.

Design Element 28-4: Detailed Two-Channel Monopulse Receiver for Range Track

The design approach for implementation of this design requirement will be discussed in detail in the classified addendum.

Design Element 28-5: Detect ECM and Apply ECCM for Range Track

The design approach for implementation of this design requirement will be discussed in detail in the classified addendum

2.28.3 Functional Element Software Design

.This section describes the software design necessary to implement the functional element requirements for range tracking as outlined in section 2.28.1 and the design approach as outlined in section 2.28.2 Sections 2.28.3 is organized as follows. Following a general discussion the section has four parts. The first part describes the overall subroutine hierarchy and gives capsule descriptions of the relevant subroutines; the second part contains the logical flow for the functional element as a whole, and describes the major operations represented by each block in the flow chart; the third part presents detailed flow charts for the subroutines; and the last part contains a description of all input and output for the functional element as a whole and for each subroutine that implements a significant part of the functional element.

In ESAMS 2.7, there are in principle six different implementations of this design requirements, corresponding to the six different non-zero values of IRSTYP(2). Table 2.28-1 shows how these values (representing 6 different servo types) are used in these six cases.

TABLE 2.28-1. Range Servo Index IRSTYP as Director through the Range Track FE.

IRSTYP	**Use In**	
	INITAG	**UPD8AG**
0	Not used. No trap on its use.	Not used. Trapped with a “fatal” error message.
1	Call RNGTRI to initialize servo SVORNG (same as 2)	Call SVORNG (same as 2)
2	Call RNGTRI to initialize servo SVORNG (same as 1)	Call SVORNG (same as 1)
3	Call INIRGA to initialize servo RNGSVA.	Call RNGSVA for tracker servo.
4	Call RGHKI to initialize g-h-k filter RGHKF.	Call RGHKF for g-h-k range track filtering.
5	Call INIRSA to initialize servo SVORSA.	Call SVORSA for tracker servo.
6	Call RGHKI to initialize g-h filter RGHF.	Call RGHF for g-h range-track filtering.
7	Not used. No trap on its use.	Not used. Trapped with a “fatal” error message.
8	Not used. No trap on its use.	Not used. Trapped with a “fatal” error message.
9	Call RNGSV1 to initialize itself.	Call RNGSV1 for tracker servo.
Other	Not used. No trap on its use.	Not used. Trapped with a “fatal” error message.

Only the design approach for implementing the design requirement for digital range track filters is discussed here. A good reference for track filters is [38], chapter 2. The most general track filter method used in ESAMS 2.7 is called the “g-h-k” filter; it is also known—perhaps more widely—as the “$\alpha-\beta-\gamma$” filter, which is the name used in [38]. This filter builds and maintains a smoothed record of a coordinate x, its velocity (i.e., time-rate) v, and its acceleration a over a sequence of discrete sample times, and it predicts a new coordinate value one sample time in the future from the smoothed x, v, and a. The filtered coordinate can be any coordinate of interest; here it will be the range coordinate. This filter is the type in DEs 28-2 and 3.

As stated earlier, the monopulse trackers use the waveform-driven model of waveform generation [37] and [39]. In this model, there are three levels of waveform organization: (1) the lowest level is the coherent processing interval (CPI) in which pulse repetition interval, pulse width, carrier frequency, etc. are held constant; (2) one or more CPIs make up the next level, the sequence; and (3) one or more sequences make up the highest level, the group. There can be more than one group, and groups repeat throughout the engagement. The pulse-to-pulse processing is done at the CPI level. When a group begins, the tracker CPI executive routine, WFTCPI, begins accumulating running a mean of the range error output from the receiver subelement; this running mean is kept in the variable WRNGER in common block WFTDAR (Include file ‘WFTDAT’). The rane error is relative to the track range value specified by the range-gate setting, RGATES, at the start of the group. At the end of the current group (call it sample n), the final value of WRNGER is the mean range error for this sample. The tracker group executive routine, WFTGRP, calls UPD8RG with this error to update the range track (i.e., RGATES) for the current sample n. As outlined in Table 3.4-1, UPD8RG in turn calls a specific range servo or range track filter, depending on the value of IRSTYP, to perform the track update. These routines receive the group duration time GRPDUR, and the mean error for sample n, WRNGER. The requirements currently are written for the case IRSTYP = 4, so UPD8RG calls the g-h-k range-track filter RGHKF. Other filters will be added at a later update.

Range Tracking Subroutine Hierarchy

Figure 2.28-4 illustrates the call tree for the over-all range track structure implemented in ESAMS 2.7. The diagram depicts the entire model structure from the top level ZINGER (the Main subprogram) through all the routines allocated to the functional element. The subroutines allocated to the functional element are listed in Table 2.28-2, together with brief descriptions of them as related to the functional element.

Initialization is performed by the waveform tracking phase initializer, WFTINI, which calls the range track initializer INITRG. There are six groups of initialization in INITRG, with the group that is actually performed in a given run determined by the specific value of the range servo type index IRSTYP; see Table 2.28-1.

There are two groups of monopulse receiver routines called by the tracking waveform coherent processing interval (i.e., pulse-by-pulse) executive routine WFTCPI: (1) For the general three-channel and two-channel monopulse receivers, there are the routines RNGDSC, PRTION (and all below it), and INTGRT; and (2) for the special detailed two-channel monopulse receiver and electronic counter-countermeasures (ECCM) particularly associated with it, there are the additional routines RCHOP (under WFTCPI), AMLOGA, PWRDBM, and AMDETR (under INTGRT), and ARNGTB (under RNGDSC). The second group is discussed in the classified addendum. Figure 2.28-4 also shows where the range-discriminator subtree (RNGDSC) occurs in the missile seeker portion of the call tree, although this occurrence is not considered part of the range track functional element. In addition, figure 2.28-4 shows where the range-gate partition subtree (PRTION) and range-gate integration (INTGRT) subtree appear in the range-track tree for the TWS radars (under TWSEXC); although, again, these occurrences are not considered part of the range track functional element.

The tracking waveform group executive routine WFTGRP calls the range track update executive UPD8RG to update range track in response to the range error output by the receiver functions. In parallel with the initializer INITRG, UPD8RG calls one of six groups of update functions, with the group that is actually performed in a given run determined by the specific value of the range servo type index IRSTYP; see table 2.28-1. General g-h-k track filter routine GHK is shown here, although it is called not only by the range track routine RGHKF, but also by the angle and doppler track filters. Similarly, the general g-h track filter routine GH is shown, although it is called not only by the range track routines RGHF, but also by the angle and doppler track filters. In addition, WFTGRP calls WFTCCM, which calls certain routines (RDOTVA, RGGCHK, and ARGPOA, as well as INITRG) to implement special receiver ECCM measures (related to range track) at the waveform group level.

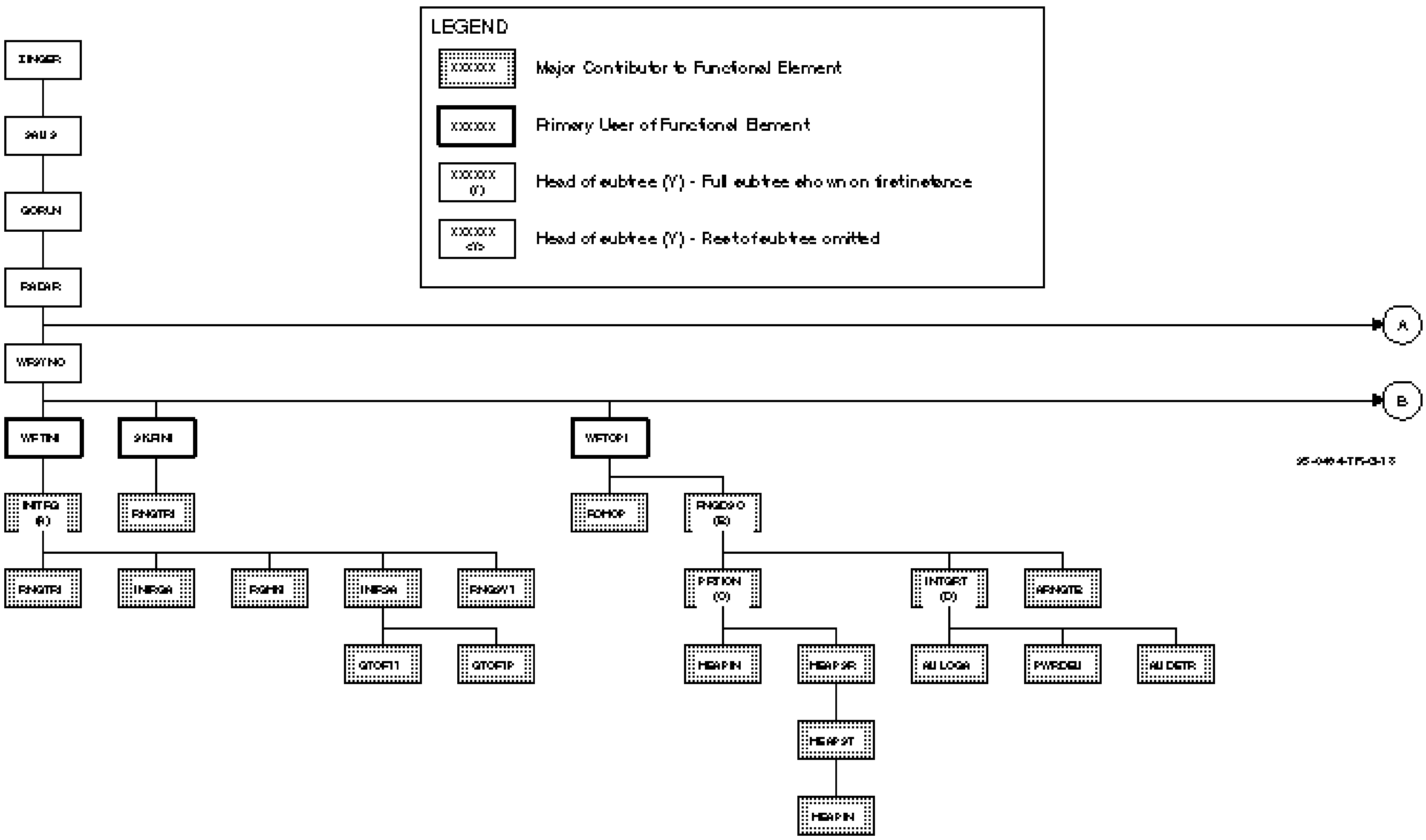

FIGURE 2.28-4. Range Track Call Tree.

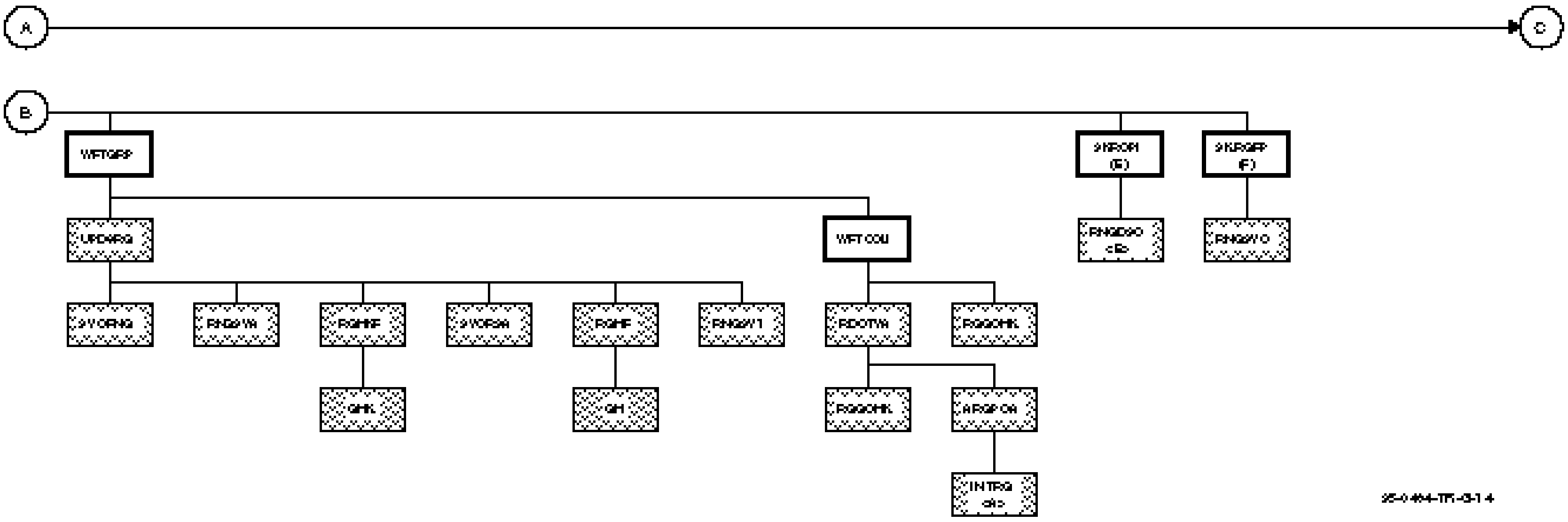

FIGURE 2.28-4. Range Track Call Tree. (Contd.)

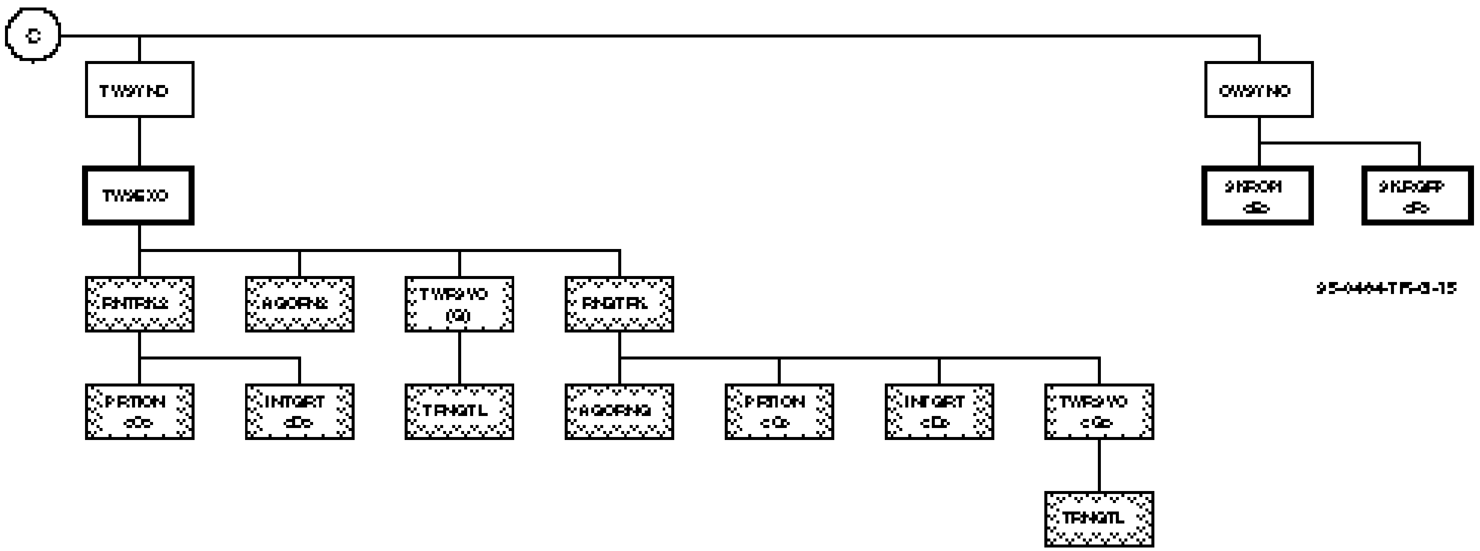

FIGURE 2.28-4. Range Track Call Tree. (Contd.)

TABLE 2.28-2. Module Descriptions for Range Tracking FE.

Module Name	Description
RNGDSC	Perform the function of producing the range error from the receiver. Called by WFTCPI, the CPI-level of waveform processing for track.
PRTION	Partition the range gate into intervals bounded by the edges of all signal pulses with some part in the range gate. Called by RNGDSC.
HEAPIN	Used by PRTION to insert a pulse edge into the list (a heap) of edges found. Called by HEAPST as well as PRTION.
HEAPSR	Sorts a list array arranged as a heap. Called by PRTION.
HEAPST	Rearranges a list array in the form of a heap. Called by HEAPSR.
INTGRT	Integrates the signals in the early and late gates of a split gate range tracker. Called by RNGDSC.
AMLOGA	Performs log amplifier function for special detailed two-channel monopulse receiver. Called by INTGRT.
PWRDBM	Function to return output power in dBw from input voltage, for special detailed receiver. Called by INTGRT.
AMDETR	Performs range detector function for special detailed two-channel monopulse receiver. Called by INTGRT.
RCHOP	Applies special range-track ECCM. Called by WFTCPI.
ARNGTB	Function to return output for range error for special detailed two-channel monopulse receiver. Called by RNGDSC.
UPD8RG	Executive routine for range servo or range track filtering. Calls specific range track servo or filter routines according to the range servo type index IRSTYP. Called by WFTGRP, the group-level of waveform processing for track.
SVORNG	Performs generic range servo function for the default range servo type IRSTYP = 1 and 2. Called by UPD8RG.
RNGSVA	Performs detailed range servo functions for the range servo type IRSTYP = 3. Called by UPD8RG.
RGHKF	Performs g-h-k range track filtering. Called by UPD8RG.
GHK	Performs generic g-h-k track filtering. Called by RGHKF.
SVORSA	Performs detailed range servo functions for the range servo type IRSTYP = 5. Called by UPD8RG.
RGHF	Performs g-h range track filtering. Called by UPD8RG.
GH	Performs generic g-h track filtering. Called by RGHF.
RNGSV1	Performs special detailed range servo functions for the range servo type IRSTYP = 9. Called by UPD8RG.
RDOTVA	Performs range-rate/doppler comparison to support ECCM logic. Called by WFTCCM.
RGGCHK	Performs range guard-gate check to support ECCM logic. Called by WFTCCM and RDOTVA.
ARGPOA	Performs anti-velocity-gate pulloff logic to support ECCM logic. Called by RDOTVA.
INITRG	Executive routine for initialization of range track servo or filter functions. Called by WFTINI, the initialization executive for the CPI-level of waveform processing for track.
RNGTRI	Initializes the difference equation coefficients for generic servo routine SVORNG. Called by INITRG.
INIRGA	Initializes the difference equation coefficients and other parameters for detailed servo routine RNGSVA. Called by INITRG.
RGHKI	Initializes the g-h-k range track filter quantities in common RGHK for filter routine RGHKF; also initializes for the g-h filter routine RGHF. Called by INITRG.

TABLE 2.28-2. Module Descriptions for Range Tracking FE. (Contd.)

Module Name	Description
INIRSA	Initializes the difference equation coefficients and other parameters for detailed servo routine SVORSA. Called by INITRG.
GTCF11	Computes difference equation coefficients for a filter with one zero and one pole. Called by INIRSA.
GTCF1P	Computes difference equation coefficients for a filter with one pole. Called by INIRSA.

Functional and Data Flow Diagrams

Figure 2.28-5 shows the top-level logical flow of the monopulse range track implementation. Subprogram names appear in parentheses at the bottom of each process block. The numbered blocks are described below.

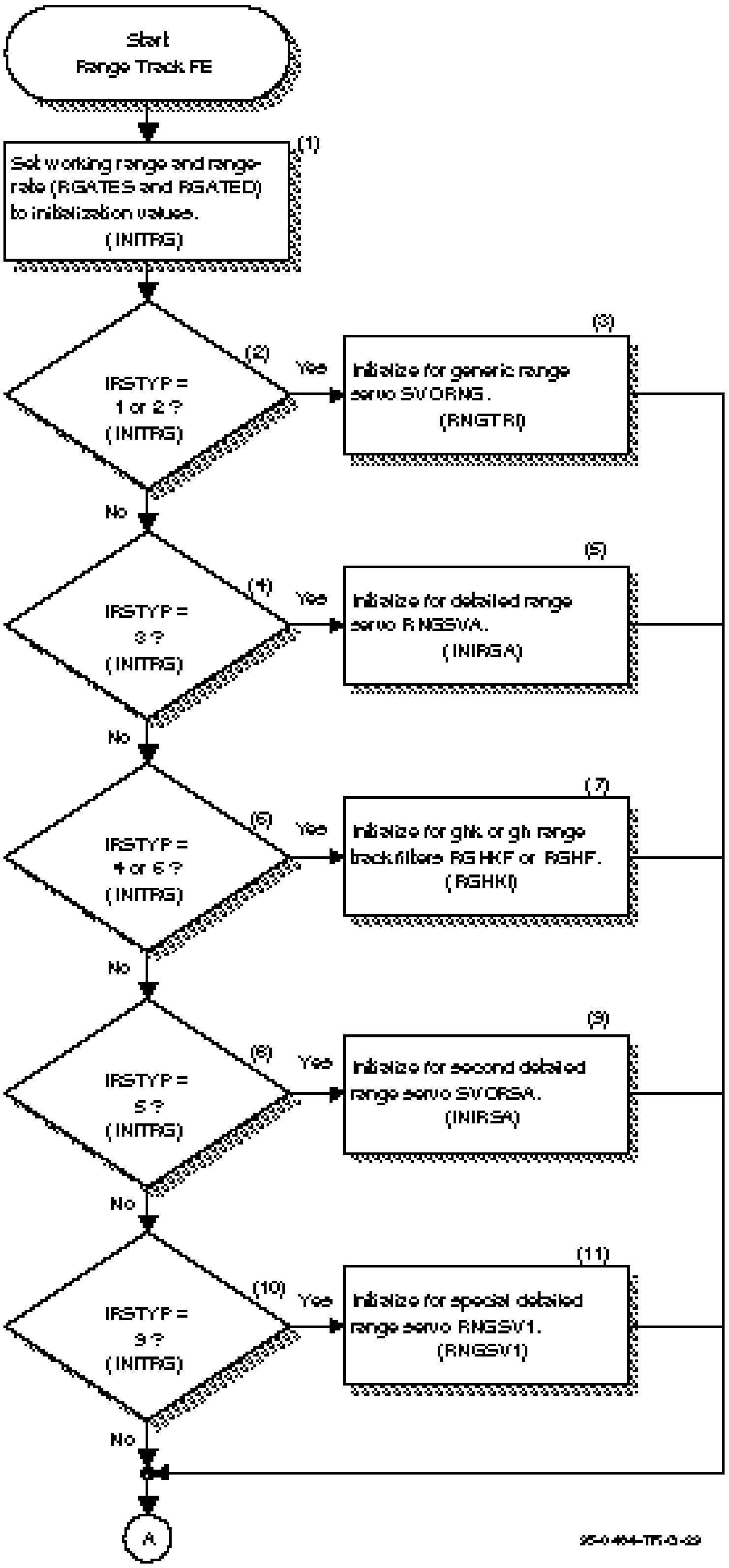
Start
Range Track FE
(1)
Set working range and range-rate (RGATES and RGATED) to initialization values.
(INITRG)
(2)
IRSTYP = 1 or 2 ? (INITRG)
Yes
(3)
Initialize for generic range servo SVORNG.
(RNGTRI)
No
(4)
IRSTYP = 3 ? (INITRG)
Yes
(5)
Initialize for detailed range servo RNGSVA.
(INIRGA)
No
(6)
IRSTYP = 4 or 6 ? (INITRG)
Yes
(7)
Initialize for ghk or gh range track filters RGHKF or RGHF.
(RGHKI)
No
(8)
IRSTYP = 5 ? (INITRG)
Yes
(9)
Initialize for second detailed range servo SVORSA.
(INIRSA)
No
(10)
IRSTYP = 9 ? (INITRG)
Yes
(11)
Initialize for special detailed range servo RNGSV1.
(RNGSV1)
No
A
95-0404-TR-G-22

FIGURE 2.28-5. Monopulse Range Track Functional and Data Flow Diagram.

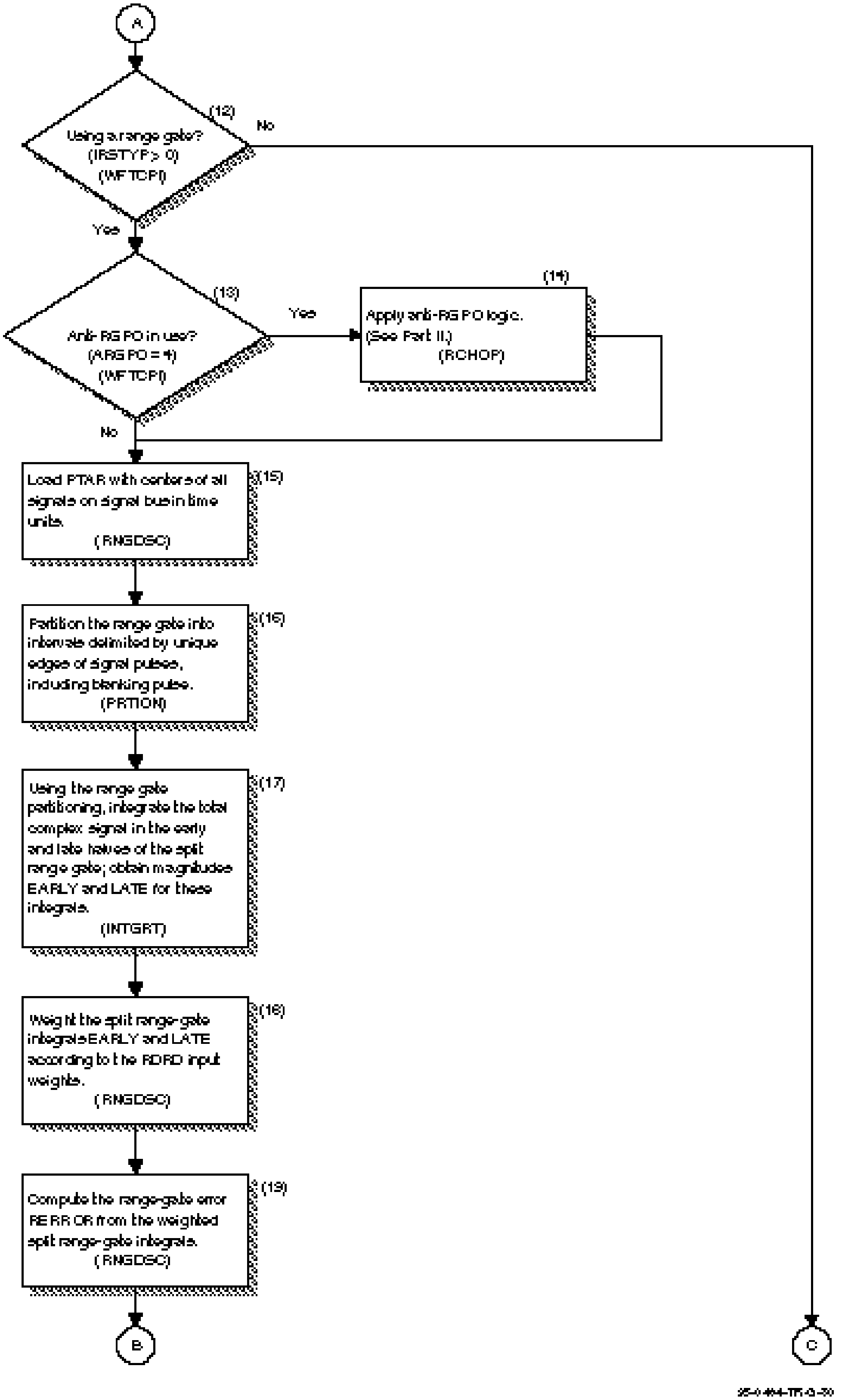
A
(12)
Using a range gate?
(IRSTYP > 0)
(WFTCPI)
No
Yes
(13)
Anti-RGPO in use?
(ARGPO = 4)
(WFTCPI)
Yes
(14)
Apply anti-RGPO logic.
(See Part II.)
(RCHOP)
No
(15)
Load PTAR with centers of all signals on signal bus in time units.
(RNGDSC)
(16)
Partition the range gate into intervals delimited by unique edges of signal pulses, including blanking pulse.
(PRTION)
(17)
Using the range gate partitioning, integrate the total complex signal in the early and late halves of the split range gate; obtain magnitudes EARLY and LATE for these integrals.
(INTGRT)
(18)
Weight the split range-gate integrals EARLY and LATE according to the RDRD input weights.
(RNGDSC)
(19)
Compute the range-gate error RERROR from the weighted split range-gate integrals.
(RNGDSC)
B
C
95-0494-TR-G-30

FIGURE 2.28-5. Monopulse Range Track Functional and Data Flow Diagram. (Contd.)

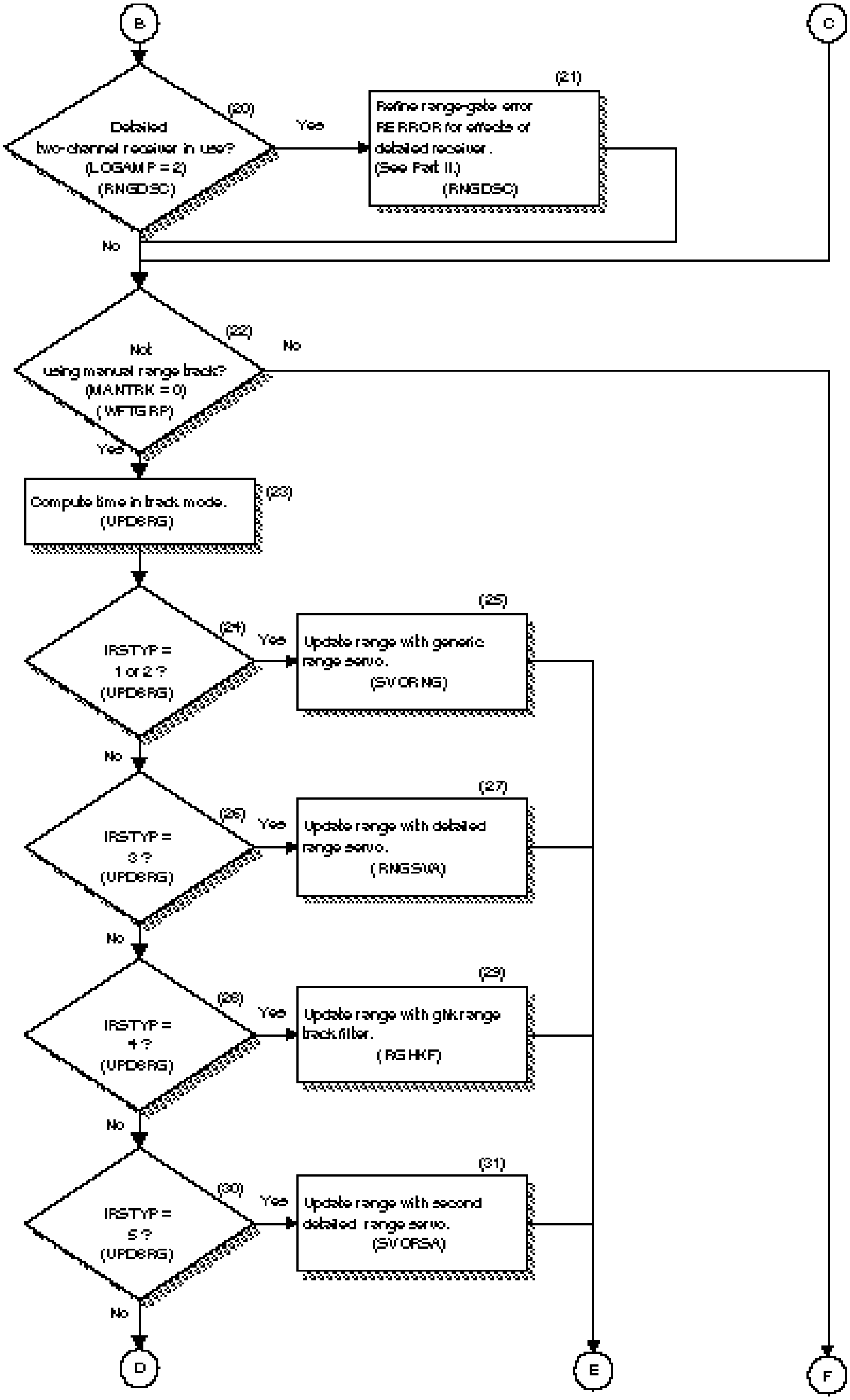

FIGURE 2.28-5. Monopulse Range Track Functional and Data Flow Diagram. (Contd.)

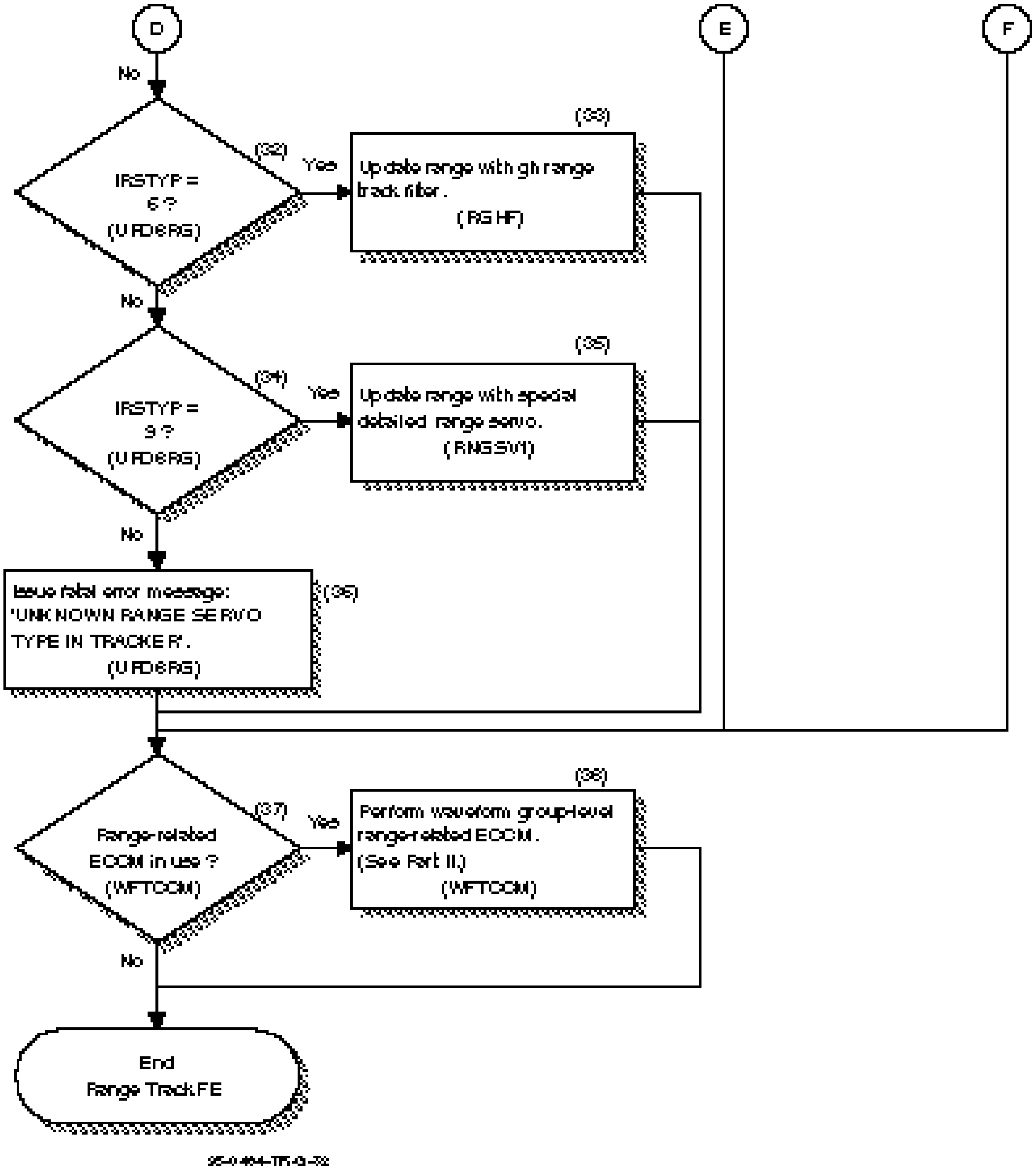
D
E
F
No
IRSTYP = 6 ? (UPD8RG)
(32)
Yes
(33)
Update range with g-h range track filter. (RGHF)
No
IRSTYP = 9 ? (UPD8RG)
(34)
Yes
(35)
Update range with special detailed range servo. (RNGSV1)
No
Issue fatal error message: 'UNKNOWN RANGE SERVO TYPE IN TRACKER'. (UPD8RG)
(36)
Range-related ECCM in use ? (WFTCCM)
(37)
Yes
(38)
Perform waveform group-level range-related ECCM. (See Part II.) (WFTCCM)
No
End Range TrackFE
95-0404-TR-G-32

FIGURE 2.28-5. Monopulse Range Track Functional and Data Flow Diagram. (Contd.)

<u>Block 1</u>. The range track initialization executive routine INITRG sets the working values of range-gate settings RGATES and range-rate RGATED to track-truth values for initialization.

<u>Blocks 2, 4, 6, 8, 10</u>. The range track initialization executive routine INITRG tests the range servo type index IRSTYP for the values 1, 2, 3, 4, 5, 6, or 9, and calls the appropriate initialization routine for range servo or range track filter. If IRSTYP is not in the range given above, INITRG does nothing; however, such out-of-range values are trapped in routine UPD8RG discussed below.

Block 3. For IRSTYP = 1 or 2, INITRG calls RNGTRI, which initializes coefficients used by the generic range servo implemented as routine SVORNG. These are generic type I or II servos.

Block 5. For IRSTYP = 3, INITRG calls the routine INIRGA, which initializes variables for the detailed range servo routine RNGSVA.

Block 7. For IRSTYP = 4 or 6, INITRG calls the routine RGHKI, which initializes variables for the range track ghk or gh filter routines RGHKF and RGHF, respectively.

Block 9. For IRSTYP = 5, INITRG calls the routine INIRSA, which initializes variables for the second detailed range servo routine SVORSA.

Block 11. For IRSTYP = 9, INITRG calls the routine RNGSV1, which initializes variables for itself; it is the special detailed range servo associated with the SMART project.

Block 12. WFTCPI tests to see if a range gate for tracking is in use (i.e., IRSTYP > 0); if not so, it bypasses range-gate processing at the waveform CPI level.

Block 13. In range-gate processing, WFTCPI tests the option switch ARGPO to see if anti-range-gate pulloff ECCM is in use.

Block 14. If ARGPO = 4, WFTCPI calls RCHOP to perform anti-RGPO functions for the waveform CPI level. (See Part II.)

Block 15. The range error executive discriminator routine, RNGDSC, loads the pulse time array PTAR with the centers of all signal pulses on the signal bus.

Block 16. The range-gate partitioning routine PRTION considers all signals on the signal bus, including the blanking pulse, and determines all unique edges of signal pulses. The leading and trailing edge of each signal pulse is loaded into the pulse time array PTAR, and an ordered list of all unique edges of signals is returned in the array TRGEP, with the number of such edges found being NEP.

Block 17. Routine INTGRT performs the integration of signals over the early and late halves of the split range gate, using the partition information developed by PRTION, and taking account of the blanking pulse, if present. The magnitude of the signal integral over the early half of the range gate is returned as the value EARLY, and that over the late half is returned as LATE.

Block 18. RNGDSC applies the RDRD input weights for the split rang-gate, WTERLY and WTLATE, to the early and late gate integral magnitudes.

Block 19. RNGDSC then computes the range error, RERROR, by computing the ratio of the difference of the weighted early and late gate integrals to the sum, scaling this by half the range-gate width in time, and converting this time value into its range equivalent in distance.

Block 20. RNGDSC tests the detailed receiver option flag, LOGAMP, to see if the detailed receiver is in use.

Block 21. If LOGAMP = 2, RNGDSC performs computations to refine the range error, RERROR, for the effects of the detailed receiver. (See Part II.)

Block 22. The waveform group-level executive tests the option switch MANTRK to see if automatic tracking or manual tracking is to be done. If MANTRK = 0, the automatic tracking covered by this VSDR is performed by calling UPD8RG; otherwise, the manual tracking option is followed.

Block 23. UPD8RG computes the current time in the track mode.

Blocks 24, 26, 28, 30, 32, and 34. The range servo and range track filter executive routine UPD8RG tests the range servo type index IRSTYP for the values of 1, 2, 3, 4, 5, 6, and 9, and calls the relevant range servo or range track filter routine for range tracking. UPD8RG is called at the group-level of waveform processing for track by WFTGRP.

Block 25. For IRSTYP = 1 or 2, UPD8RG calls the generic range servo routine SVORNG to update the range-gate setting. This routine implements both type I and II generic servos.

Block 27. For IRSTYP = 3, the detailed servo routine RNGSVA to updates the range-gate setting.

Block 29. RGHKF performs the range track ghk filter simulation to update the range-gate setting appropriate for IRSTYP = 4.

Block 31. SVORSA performs the range servo simulation to update the range-gate setting appropriate for IRSTYP = 5.

Block 33. RGHF performs the range track gh filter simulation to update the range-gate setting appropriate for IRSTYP = 6.

Block 35. RNGSV1 performs the range servo simulation to update the range-gate setting appropriate for IRSTYP = 9. (See Part II)

Block 36. When IRSTYP is out of range (i.e., not in the set of (1, 2, 3, 4, 5, 6, 9)), UPD8RG issues a fatal error message.

Block 37. Waveform group-level tracking ECCM executive WFTCCM (called by WFTGRP) tests various flags and switches to see if range-related ECCM is appropriate. (See Part II.)

Block 38. WFTCCM directs range-track-related ECCM functions as appropriate. (See Part II.)

Subroutine Flow Charts

Functional flow diagrams of the subroutines which support the Range Track functional element, in the example of digital track update discussed here, are shown in figures 2.28-6 through 13.

Figures 2.28-6,7,8, and 9 deal with the subelement of range error measurement by the receiver. Figure 2.28-6 shows the top routine for this element, RNGDSC. Figures 2.28-7 and 8 show the partitioning routine PRTION and the three heap management routines that support it. Figure 2.28-9 shows the routine that actually performs the integration over the early and late gates of the split-gate tracker, INTGRT.

Figure 2.28-10 shows the range correction subelement top level routines for initialization, INITRG, and for track range update, UPD8RG. Figure 2.28-11 shows the specific initialization routine for the ghk range track filter, RGHKI. Figure 2.28-12 shows the specific routine for the ghk range track filter update, RGHKF. Figure 2.28-13 shows the generic routine for ghk track filter update, GHK; this routine is shown here for completeness, although it is also used in the angle and doppler track functional elements.

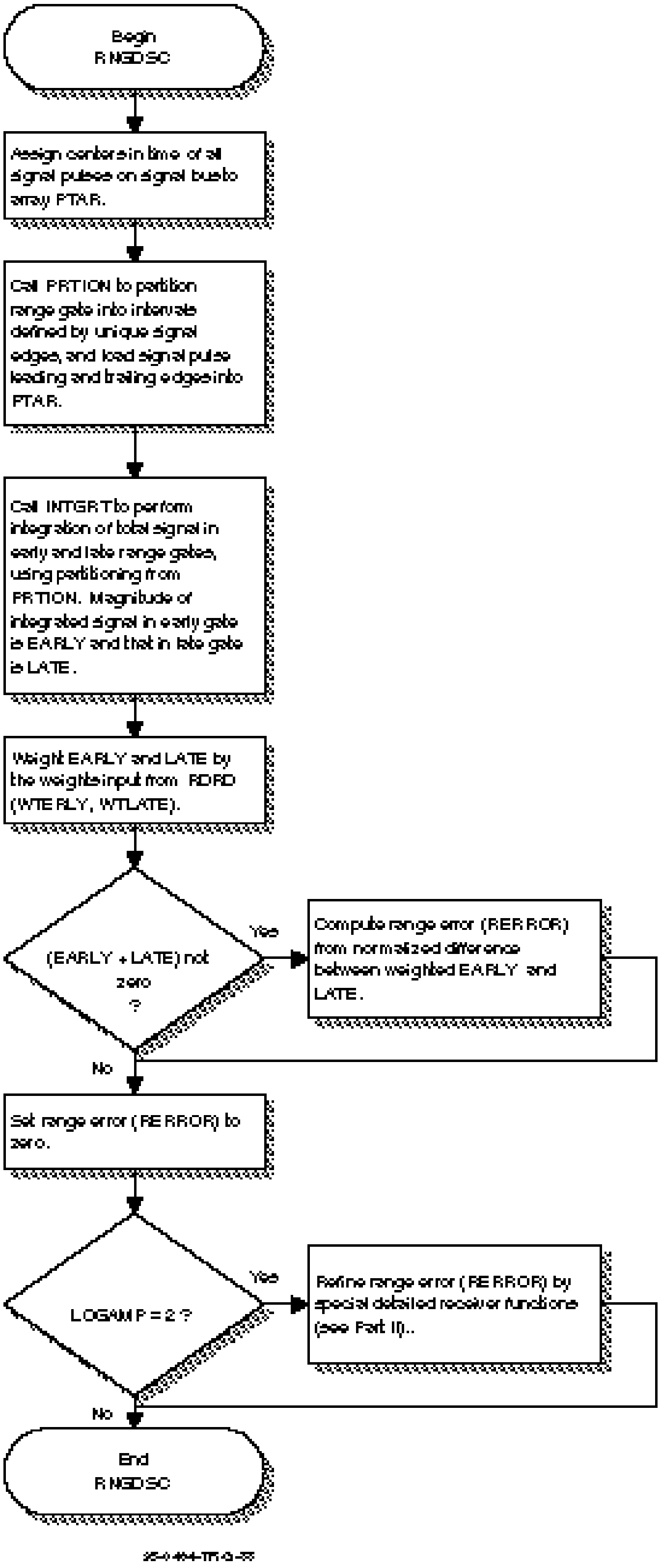
Begin
RNGDSC
Assign centers in time of all signal pulses on signal bus to array PTAR.
Call PRTION to partition range gate into intervals defined by unique signal edges, and load signal pulse leading and trailing edges into PTAR.
Call INTGRT to perform integration of total signal in early and late range gates, using partitioning from PRTION. Magnitude of integrated signal in early gate is EARLY and that in late gate is LATE.
Weight EARLY and LATE by the weights input from RDRD (WTERLY, WTLATE).
(EARLY + LATE) not zero ?
Yes
Compute range error (RERROR) from normalized difference between weighted EARLY and LATE.
No
Set range error (RERROR) to zero.
LOGAMP = 2 ?
Yes
Refine range error (RERROR) by special detailed receiver functions (see Part II)..
No
End
RNGDSC
95-0404-TF-G-77

FIGURE 2.28-6. Functional Flow Diagram for Subroutine RNGDSC.

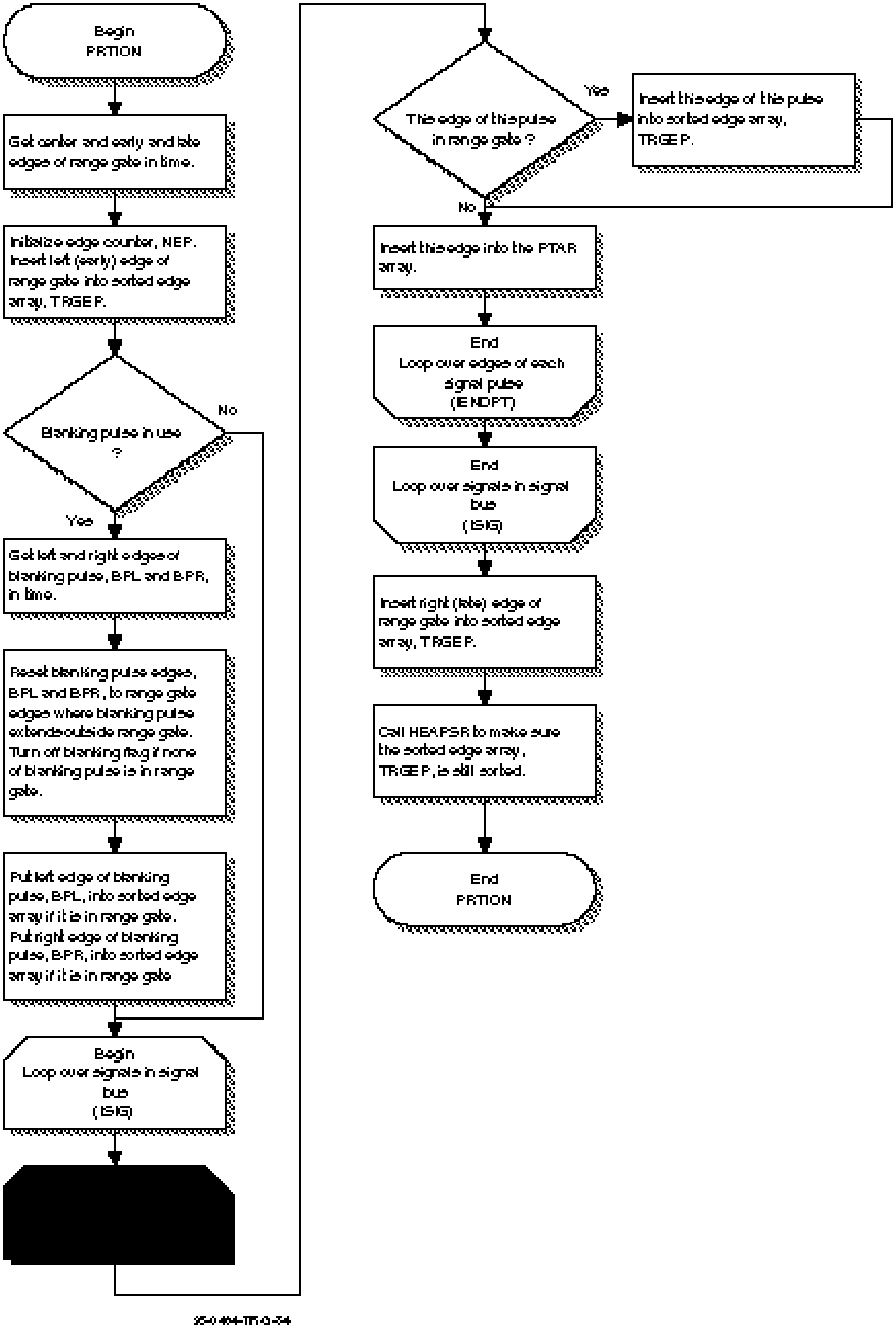
Begin
PRTION
Get center and early and late edges of range gate in time.
Initialize edge counter, NEP.
Insert left (early) edge of range gate into sorted edge array, TRGEP.
Blanking pulse in use ?
No
Yes
Get left and right edges of blanking pulse, BPL and BPR, in time.
Reset blanking pulse edges, BPL and BPR, to range gate edges where blanking pulse extends outside range gate.
Turn off blanking flag if none of blanking pulse is in range gate.
Put left edge of blanking pulse, BPL, into sorted edge array if it is in range gate.
Put right edge of blanking pulse, BPR, into sorted edge array if it is in range gate
Begin
Loop over signals in signal bus
(ISIG)
[redacted]
This edge of this pulse in range gate ?
Yes
No
Insert this edge of this pulse into sorted edge array, TRGEP.
Insert this edge into the PTAR array.
End
Loop over edges of each signal pulse
(IENDPT)
End
Loop over signals in signal bus
(ISIG)
Insert right (late) edge of range gate into sorted edge array, TRGEP.
Call HEAPSR to make sure the sorted edge array, TRGEP, is still sorted.
End
PRTION
95-0464-TR-G-34

FIGURE 2.28-7. Functional Flow Diagram for Subroutine PRTION.

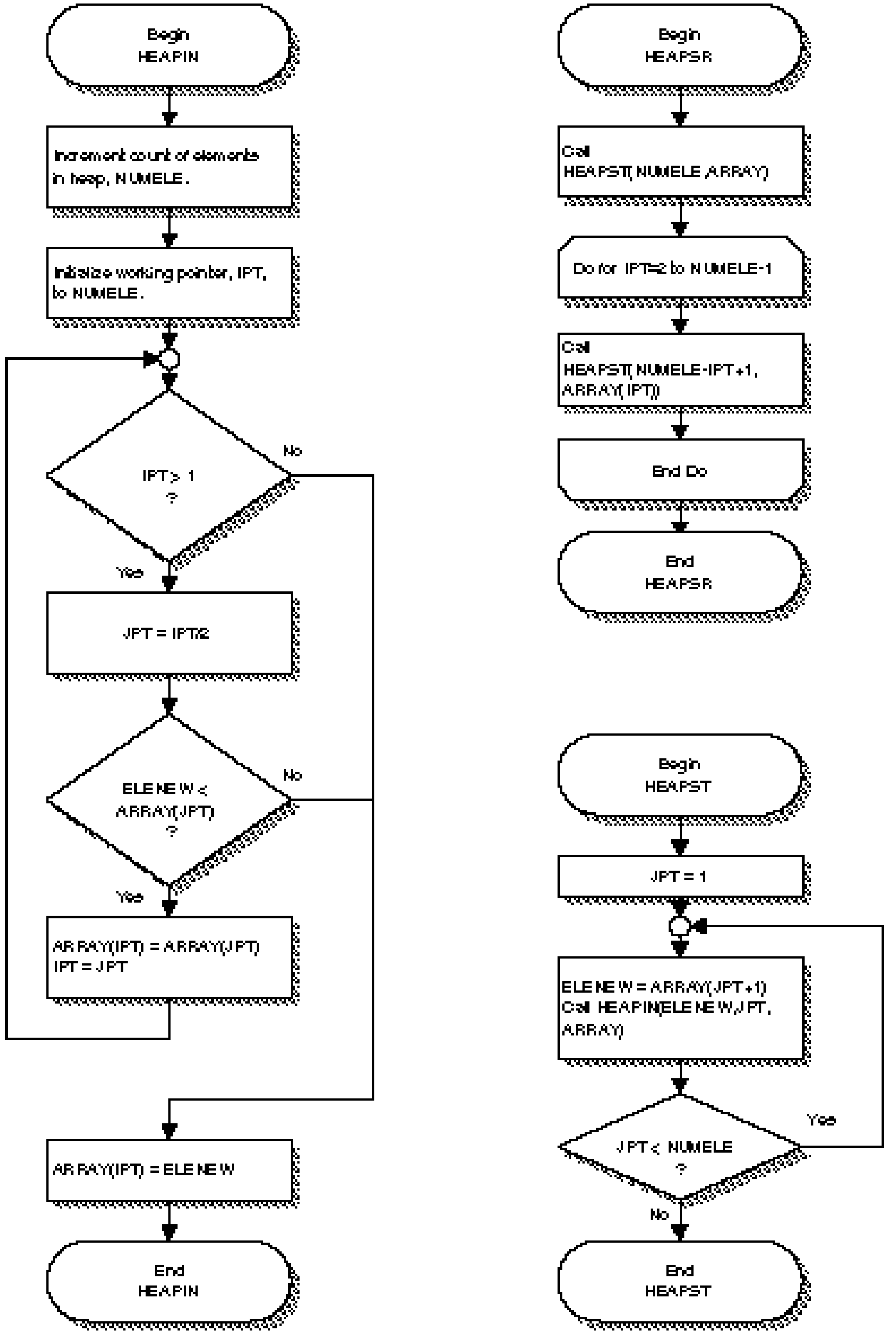

FIGURE 2.28-8. Functional Flow Diagram for Subroutines HEAPIN, HEAPSR, and HEAPST.

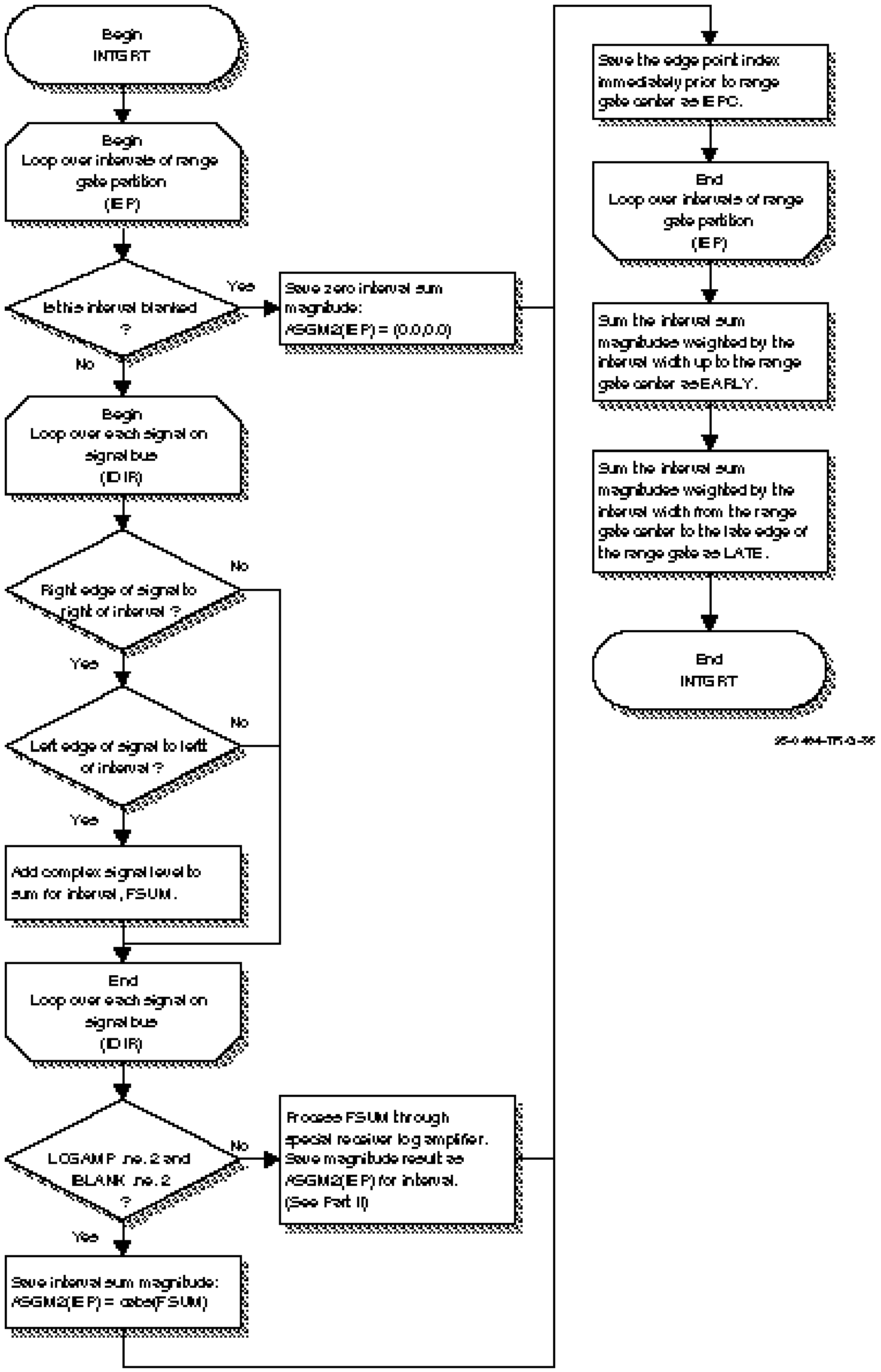
Begin
INTGRT
Begin
Loop over intervals of range gate partition
(IEP)
Is this interval blanked ?
Yes
Save zero interval sum magnitude:
ASGM2(IEP) = (0.0,0.0)
No
Begin
Loop over each signal on signal bus
(IDIR)
Right edge of signal to right of interval ?
No
Yes
Left edge of signal to left of interval ?
No
Yes
Add complex signal level to sum for interval, FSUM.
End
Loop over each signal on signal bus
(IDIR)
LOGAMP .ne. 2 and IBLANK .ne. 2
?
No
Process FSUM through special receiver log amplifier. Save magnitude result as ASGM2(IEP) for interval. (See Part II)
Yes
Save interval sum magnitude:
ASGM2(IEP) = cabs(FSUM)
Save the edge point index immediately prior to range gate center as IEPC.
End
Loop over intervals of range gate partition
(IEP)
Sum the interval sum magnitudes weighted by the interval width up to the range gate center as EARLY.
Sum the interval sum magnitudes weighted by the interval width from the range gate center to the late edge of the range gate as LATE.
End
INTGRT
95-0464-TF-G-26

FIGURE 2.28-9. Functional Flow Diagram for Subroutine INTGRT.

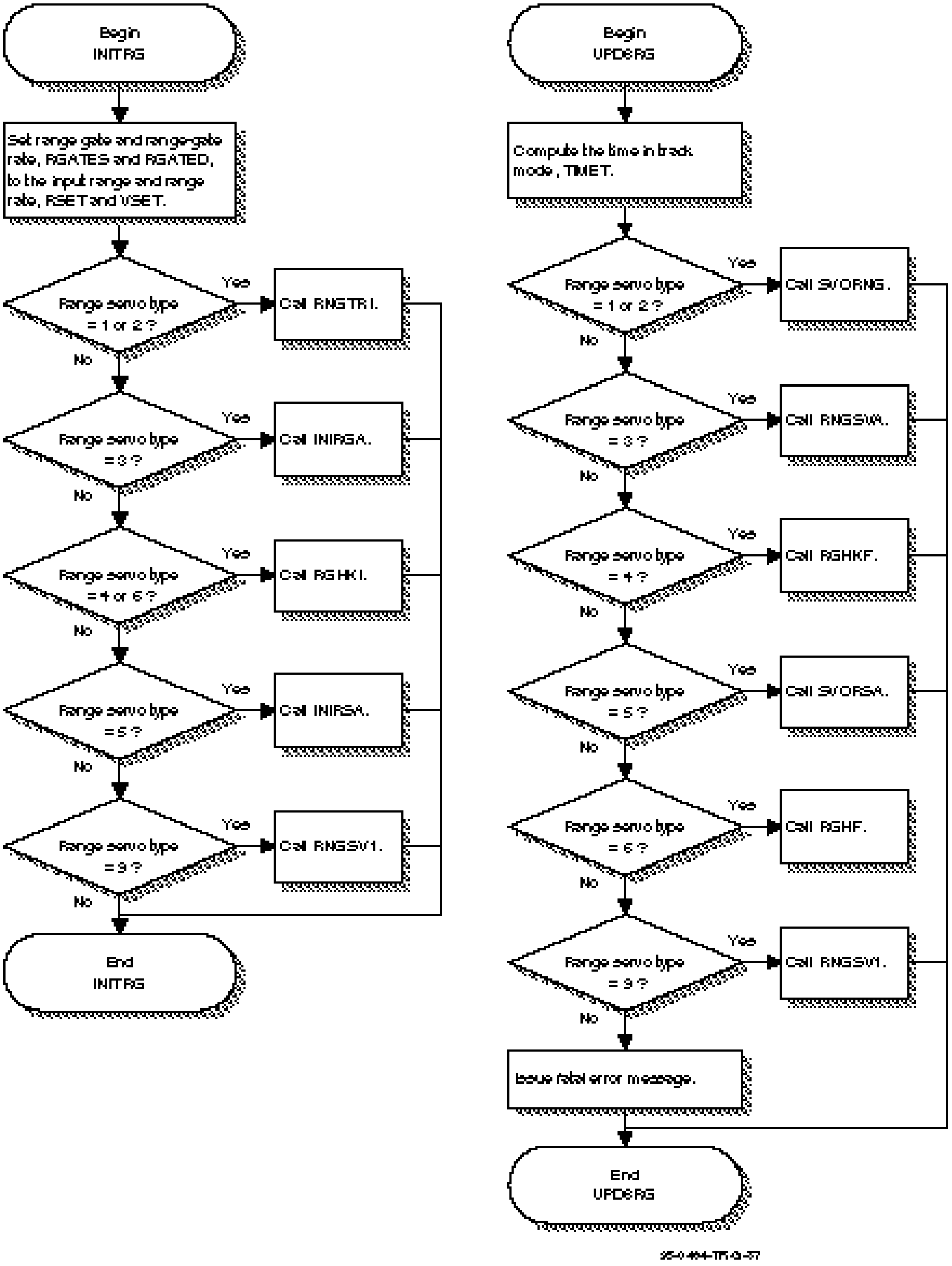

FIGURE 2.28-10. Functional Flow Diagram for Subroutines INITRG and UPD8RG.

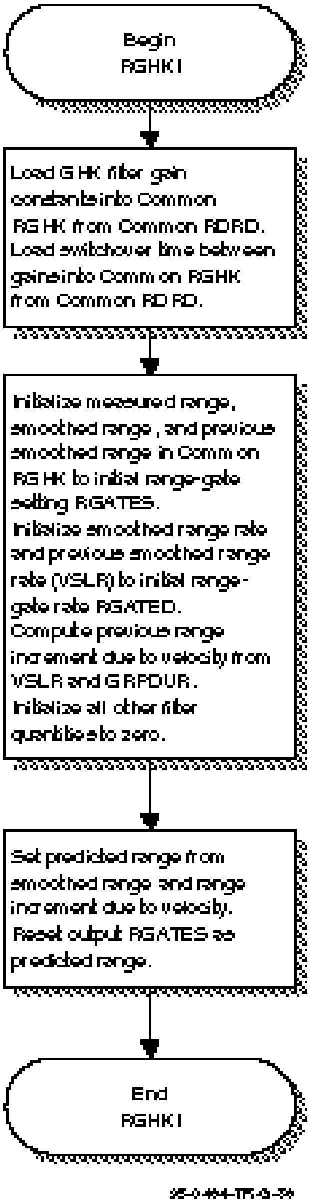
Begin
RGHKI
Load GHK filter gain constants into Common RGHK from Common RDRD. Load switchover time between gains into Comm on RGHK from Common RDRD.
Initialize measured range, smoothed range, and previous smoothed range in Comm on RGHK to initial range-gate setting RGATES. Initialize smoothed range rate and previous smoothed range rate (VSLR) to initial range-gate rate RGATED. Compute previous range increment due to velocity from VSLR and GRPDUR. Initialize all other filter quantities to zero.
Set predicted range from smoothed range and range increment due to velocity. Reset output RGATES as predicted range.
End
RGHKI
95-0494-TF-G-39

FIGURE 2.28-11. Functional Flow Diagram for Subroutine RGHKI.

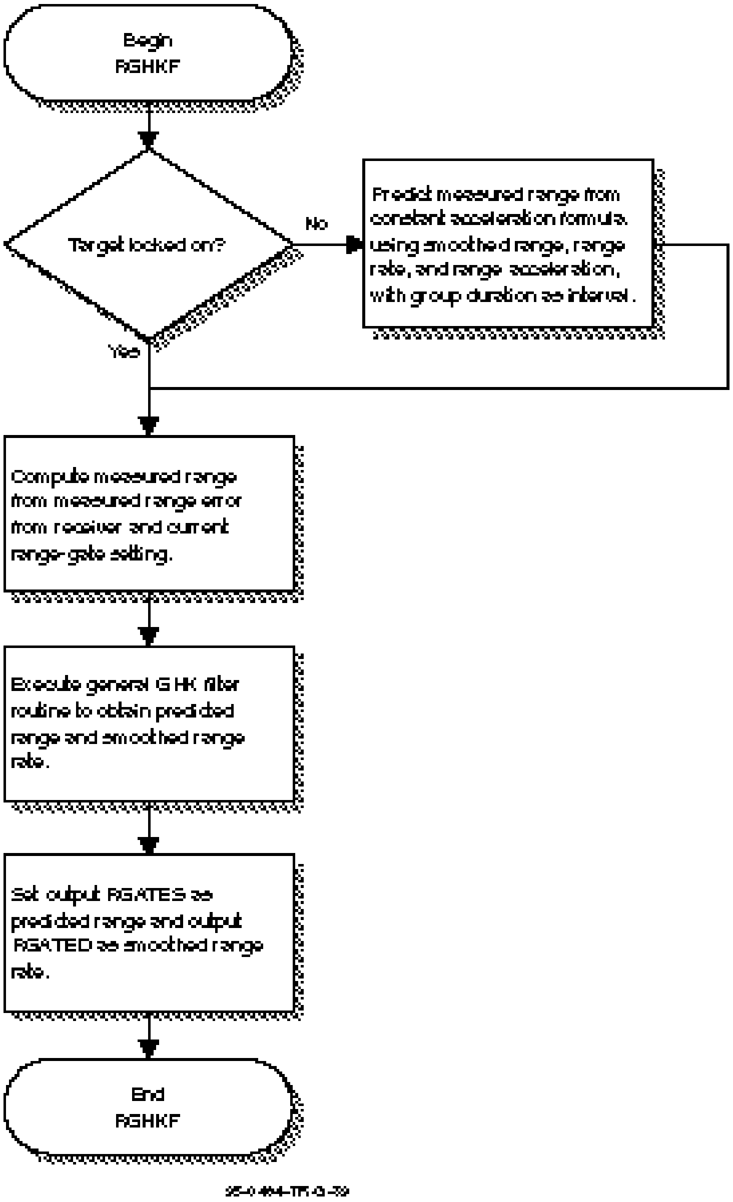
Begin
RGHKF
Target locked on?
No
Predict measured range from constant acceleration formula using smoothed range, range rate, and range acceleration, with group duration as interval.
Yes
Compute measured range from measured range error from receiver and current range-gate setting.
Execute general GHK filter routine to obtain predicted range and smoothed range rate.
Set output RGATES as predicted range and output RGATED as smoothed range rate.
End
RGHKF
95-0494-TR-G-39

FIGURE 2.28-12. Functional Flow Diagram for Subroutine RGHKF.

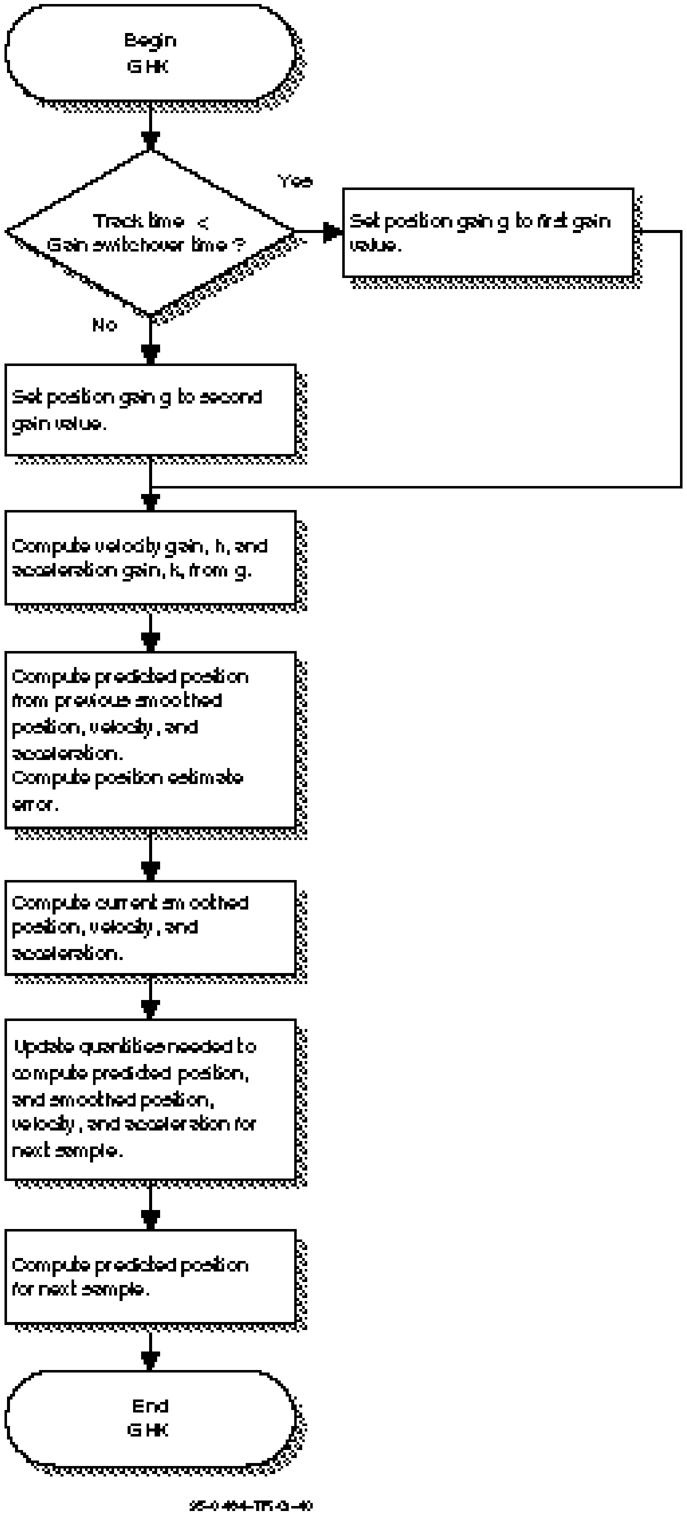
Begin
GHK
Track time <
Gain switchover time ?
Yes
Set position gain g to first gain value.
No
Set position gain g to second gain value.
Compute velocity gain, h, and acceleration gain, k, from g.
Compute predicted position from previous smoothed position, velocity, and acceleration.
Compute position estimate error.
Compute current smoothed position, velocity, and acceleration.
Update quantities needed to compute predicted position, and smoothed position, velocity, and acceleration for next sample.
Compute predicted position for next sample.
End
GHK
95-0494-TF-G-40

FIGURE 2.28-13. Functional Flow Diagram for Subroutine GHK.

Range Tracking Inputs and Outputs

The user input to the Range Track Functional Element is collected in Table 2.28-3. This table includes all user input data for all range servo type index values and all range-related ECCM options.

TABLE 2.28-3. Summary of User Input Data to Range Track.

Name	Source	Description
AMFGAN	Input in RDRD	Range servo acquisition mode filter gain (IRSTYP=3).
AMFZRO	Input in RDRD	Range servo acquisition mode filter zero (IRSTYP=3).
ARGPO(-)	Input in RDRD	Array of anti-range-gate pulloff option flags, for radar types (see Part II).
BPFRCTS(-)	Input in RDRD	Array of fractional blanking pulse widths, for radar types.
BPMINS(-)	Input in RDRD	Array of minimum signal separations for blanking pulse generation, for radar types.
COMPGN	Input in RDRD	Comparator gain (IRSTYP=5).
DISCGN	Input in RDRD	Discriminator gain (IRSTYP=5).
DISMID(-)	Input in RDRD	Array of separations between range guard-gate center and range-gate center, for radar types (see Part II).
FBFGAN	Input in RDRD	Feedback filter gain (IRSTYP=3).
FBFTAU	Input in RDRD	Feedback filter time constant (IRSTYP=3).
GRDWDT(-)	Input in RDRD	Array of range guard-gate widths, for radar types (see Part II).
GRPDUR	Input in waveform data in RDRD; retrieved from array RWFARY of Common WAVEFM	Duration time of group in waveform.
GTHRSH(-)	Input in RDRD	Array of range guard-gate alarm thresholds, for radar types (see Part II).
IRSTYP(-)	Input in RDRD; working value in Common SVOVAR	Array of range servo type indices for radar types.
LOGAMP(-)	Input in RDRD; working value in Common GRADAR	Array of flags indicating whether the special detailed two-channel monopulse receiver is to be used, for radar types.
PWTX(-)	Input in waveform data in RDRD; retrieved from array RWFARY of Common WAVEFM; current values in Common GRADAR	Array of transmitter pulse widths, for radar types.
RANGHK(-)	Input in RDRD	Array of early and late gain parameters, g, for range g-h-k track filter (IRSTYP=4 or 6).
RANTSW	Input in RDRD	Time to switch between early and late values of track filter parameter g (IRSTYP=4 or 6).
RGF1GN	Input in RDRD	Range filter no. 1 gain (IRSTYP=5).
RGF1PL	Input in RDRD	Range filter no. 1 pole (IRSTYP=5).
RGF1ZR	Input in RDRD	Range filter no. 1 zero (IRSTYP=5).
RGF2GN	Input in RDRD	Range filter no. 2 gain (IRSTYP=5).
RGF2PL	Input in RDRD	Range filter no. 2 pole (IRSTYP=5).

TABLE 2.28-3. Summary of User Input Data to Range Track. (Contd.)

Name	Source	Description
RGF2ZR	Input in RDRD	Range filter no. 2 zero (IRSTYP=5).
RGF3GN	Input in RDRD	Range filter no. 3 gain (IRSTYP=5).
RGF3PL	Input in RDRD	Range filter no. 3 pole (IRSTYP=5).
RGF3ZR	Input in RDRD	Range filter no. 3 zero (IRSTYP=5).
RGF4GN	Input in RDRD	Range filter no. 4 gain (IRSTYP=5).
RGF4PL	Input in RDRD	Range filter no. 4 pole (IRSTYP=5).
RGF4ZR	Input in RDRD	Range filter no. 4 zero (IRSTYP=5).
RGNINT	Input in RDRD	Range integrator gain (IRSTYP=3).
RKSRV(-)	Input in RDRD; working values in Common SVOVAR	Servo parameter for generic servo (IRSTYP=1,2), Locke’s ‘k’.
RMPGAN	Input in RDRD	Range amplifier gain (IRSTYP=3).
RMPTAU	Input in RDRD	Range amplifier time constant (IRSTYP=3).
RNSRV(-)	Input in RDRD; working values in Common SVOVAR	Servo parameter for generic servo (IRSTYP=1,2).
RRATLM	Input in RDRD	Range rate limit (IRSTYP=3).
RTSWDP	Input in RDRD	Time to switch mode filters (IRSTYP=3).
TMFGAN	Input in RDRD	Range servo track mode filter gain (IRSTYP=3).
TMFZRO	Input in RDRD	Range servo track mode zero gain (IRSTYP=3).
TRGW(-)	Input in waveform data in RDRD; retrieved from array RWFARY of Common WAVEFM; working values in Common SVOVAR	Array of range gate widths (in time), for radar types.
WRSV(-)	Input in RDRD; working values in Common SVOVAR	Servo parameter for generic servo (IRSTYP=1,2), Locke’s ‘w’.
WTERLY(-)	Input in RDRD; working values in Common SVOVAR	Array of weights to be used for the early gate of split-gate tracking, for radar types.
WTLATE(-)	Input in RDRD; working values in Common SVOVAR	Array of weights to be used for the late gate of split-gate tracking, for radar types.
XICHNL(-)	Input in RDRD	Array of flags indicating whether the three- or two-channel monopulse option is to be used, for radar types.

Tables of input and output for major routines implementing the range tracking FE are listed in the following tables for the routines listed below:

INITRG
RGHKI
RNGDSC
PRTION
HEAPIN
HEAPST
HEAPSR
INTGRT
UPD8RG
RGHKF
GHK

TABLE 2.28-4a. Subroutine INITRG—Input Data.

Name	Source	Description
AMFGAN	Common RDRD	Range servo acquisition mode filter gain (IRSTYP=3).
AMFZRO	Common RDRD	Range servo acquisition mode filter zero (IRSTYP=3).
COMPGN	Common RDRD	Comparator gain (IRSTYP=5).
DELTAT	Argument In	Time step for range updates. (Actual argument in call from WFTINI is GRPDUR.)
DISCGN	Common RDRD	Discriminator gain (IRSTYP=5).
FBFGAN	Common RDRD	Feedback filter gain (IRSTYP=3).
FBFTAU	Common RDRD	Feedback filter time constant (IRSTYP=3).
IRSTYP(-)	Common SVOVAR	Range servo type index.
RANGHK(-)	Common RDRD	Array of early and late gain parameters, g, for range g-h-k track filter (IRSTYP=4 or 6).
RANTSW	Common RDRD	Time to switch between early and late values of track filter parameter g (IRSTYP=4 or 6).
RGF1GN	Common RDRD	Range filter no. 1 gain (IRSTYP=5).
RGF1PL	Common RDRD	Range filter no. 1 pole (IRSTYP=5).
RGF1ZR	Common RDRD	Range filter no. 1 zero (IRSTYP=5).
RGF2GN	Common RDRD	Range filter no. 2 gain (IRSTYP=5).
RGF2PL	Common RDRD	Range filter no. 2 pole (IRSTYP=5).
RGF2ZR	Common RDRD	Range filter no. 2 zero (IRSTYP=5).
RGF3GN	Common RDRD	Range filter no. 3 gain (IRSTYP=5).
RGF3PL	Common RDRD	Range filter no. 3 pole (IRSTYP=5).
RGF3ZR	Common RDRD	Range filter no. 3 zero (IRSTYP=5).
RGF4GN	Common RDRD	Range filter no. 4 gain (IRSTYP=5).
RGF4PL	Common RDRD	Range filter no. 4 pole (IRSTYP=5).
RGF4ZR	Common RDRD	Range filter no. 4 zero (IRSTYP=5).
RGNINT	Common RDRD	Range integrator gain (IRSTYP=3).
RKSVR(-)	Common SVOVAR	Servo parameter for generic servo (IRSTYP=1,2), Locke’s ‘k’.
RMPGAN	Common RDRD	Range amplifier gain (IRSTYP=3).
RMPTAU	Common RDRD	Range amplifier time constant (IRSTYP=3).
RNSVR(-)	Common SVOVAR	Servo parameter for generic servo (IRSTYP=1,2).
RRATLM	Common RDRD	Range rate limit (IRSTYP=3).
RSET	Argument In	Range to which the range gate is to be set for initialization. (Actual argument in call from WFTINI is RGATES(ITRKR).)
RTSWDP	Common RDRD	Time to switch mode filters (IRSTYP=3).
TMFGAN	Common RDRD	Range servo track mode filter gain (IRSTYP=3).
TMFZRO	Common RDRD	Range servo track mode zero gain (IRSTYP=3).
VSET	Argument In	Range rate to be set for the range gate for initialization. (Actual argument in call from WFTINI is RRATE.)
WRSV(-)	Common SVOVAR	Servo parameter for generic servo (IRSTYP=1,2), Locke’s ‘w’.

TABLE 2.28-4b. Subroutine INITRG—Output Data.

Name	**Source**	**Description**
AMFGAN	Argument to INIRGA	Range servo acquisition mode filter gain (IRSTYP=3).
AMFZRO	Argument to INIRGA	Range servo acquisition mode filter zero (IRSTYP=3).
COMPGN	Argument to INIRSA	Comparator gain (IRSTYP=5).
DELTAT	Argument to RNGTRI, INIRGA, RGHKI, INIRSA	Time step for range updates. (Actual argument in call from WFTINI is GRPDUR.)
DISCGN	Argument to INIRSA	Discriminator gain (IRSTYP=5).
FBFGAN	Argument to INIRGA	Feedback filter gain (IRSTYP=3).
FBFTAU	Argument to INIRGA	Feedback filter time constant (IRSTYP=3).
IRSTYP(-)	Argument to RNGTRI	Range servo type index.
RANGHK(-)	Argument to RGHKI	Array of early and late gain parameters, g, for range g-h-k or g-h track filter (IRSTYP=4 or 6).
RANTSW	Argument to RGHKI	Time to switch between early and late values of track filter parameter g (IRSTYP=4 or 6).
RGATED(-)	Common GRADAR	Array of range-gate rates for all radar types.
RGATES(-)	Common GRADAR; Argument to RNGSV1	Array of range-gate settings for all radar types.
RGF1GN	Argument to INIRSA	Range filter no. 1 gain (IRSTYP=5).
RGF1PL	Argument to INIRSA	Range filter no. 1 pole (IRSTYP=5).
RGF1ZR	Argument to INIRSA	Range filter no. 1 zero (IRSTYP=5).
RGF2GN	Argument to INIRSA	Range filter no. 2 gain (IRSTYP=5).
RGF2PL	Argument to INIRSA	Range filter no. 2 pole (IRSTYP=5).
RGF2ZR	Argument to INIRSA	Range filter no. 2 zero (IRSTYP=5).
RGF3GN	Argument to INIRSA	Range filter no. 3 gain (IRSTYP=5).
RGF3PL	Argument to INIRSA	Range filter no. 3 pole (IRSTYP=5).
RGF3ZR	Argument to INIRSA	Range filter no. 3 zero (IRSTYP=5).
RGF4GN	Argument to INIRSA	Range filter no. 4 gain (IRSTYP=5).
RGF4PL	Argument to INIRSA	Range filter no. 4 pole (IRSTYP=5).
RGF4ZR	Argument to INIRSA	Range filter no. 4 zero (IRSTYP=5).
RGNINT	Argument to INIRGA	Range integrator gain (IRSTYP=3).
RKSVR(-)	Argument to RNGTRI	Servo parameter for generic servo (IRSTYP=1 or 2), Locke’s ‘k’.
RMPGAN	Argument to INIRGA	Range amplifier gain (IRSTYP=3).
RMPTAU	Argument to INIRGA	Range amplifier time constant (IRSTYP=3).
RNSVR(-)	Argument to RNGTRI	Servo parameter for generic servo (IRSTYP=1 or 2).
RRATLM	Argument to INIRGA, INIRSA	Range rate limit (IRSTYP=3 or 5).
RSET	Argument to RNGTRI, INIRGA, RGHKI, INIRSA	Range to which the range gate is to be set for initialization. (Actual argument in call from WFTINI is RGATES(ITRKR).)
RTSWDP	Argument to INIRGA	Time to switch mode filters (IRSTYP=3).
RZERO	Argument to RNGSV1	Local variable initialized to 0.0.
TMFGAN	Argument to INIRGA	Range servo track mode filter gain (IRSTYP=3).
TMFZRO	Argument to INIRGA	Range servo track mode zero gain (IRSTYP=3).
VSET	Argument to RGHKI	Range rate to be set for the range gate for initialization. (Actual argument in call from WFTINI is RRATE.)
WRSV(-)	Argument to RNGTRI	Servo parameter for generic servo (IRSTYP=1 or 2), Locke’s ‘w’.

TABLE 2.28-5a. Subroutine RGHKI—Input Data.

Name	Source	Description
GHKG(-)	Argument In	Array of early and late gains, g, for ghk range track filter.
GRPDUR	Argument In	Duration time of current group in waveform.
RGATES	Argument In	Range-gate setting for initialization of current radar type.
RRATE	Argument In	Range-gate rate for initialization of current radar type.
TWGHK	Argument In	Time for switchover between early and late gain values for g of ghk range track filter.

TABLE 2.28-5b. Subroutine RGHKI—Output Data.

Name	Source	Description
ASLR	Common RGHK	Initialization of previous value of smoothed range acceleration.
ASR	Common RGHK	Initialization of smoothed range acceleration.
DXALR	Common RGHK	Initialization of range increment due to acceleration.
DXVLR	Common RGHK	Initialization of range increment due to velocity.
EXR	Common RGHK	Initialization of range estimate error.
RGATES	Argument Out	Updated range-gate setting for current radar type, at end of initialization.
VSALR	Common RGHK	Initialization of range rate increment due to acceleration.
VSLR	Common RGHK	Initialization of previous value of smoothed range rate.
VSR	Common RGHK	Initialization of smoothed range rate.
XMR	Common RGHK	Initialized measured range for current group.
XPR	Common RGHK	Initialization of range prediction.
XSLR	Common RGHK	Initialization of previous value of smoothed range.
XSR	Common RGHK	Initialization of smoothed range.

TABLE 2.28-6a. Subroutine RNGDSC—Input Data.

Name	Source	Description
BPL	Argument In; Return from PRTION	The left (early) edge of the blanking pulse, if the blanking pulse is on. On return from PRTION, may be trimmed to left edge of range gate.
BPR	Argument In; Return from PRTION	The right (late) edge of the blanking pulse, if the blanking pulse is on. On return from PRTION, may be trimmed to right edge of range gate.
EARLY	Return from INTGRT	Magnitude of integral of all signals in early half of range gate.
IBLANK	Argument In; Return from PRTION	Flag indicating whether the blanking pulse is on (=1, on; =0, off). On return from PRTION, may be turned off.
LATE	Return from INTGRT	Magnitude of integral of all signals in late half of range gate.
LOGAMP	Argument In	Flag indication whether the special detailed receiver using log amplifiers is in use (=2, means the special receiver is in use).
NEP	Return from PRTION	Number of discrete pulse edges found in range gate.
NUMPRO	Argument In	Number of signals on signal bus.
POWERA	Argument In	Power input to the AGC for special detailed receiver using log amplifiers.

TABLE 2.28-6a. Subroutine RNGDSC—Input Data. (Contd.)

Name	Source	Description
PTAR(-,-)	Return from PRTION	Array of pulse time of arrival for left and right edges of all pulses on signal bus.
RGATE	Argument In	Location of center of range gate in distance.
RKDIFF	Argument In	Conversion factor for converting range error in time to range error in distance, for current setting of range gate width.
RTSI	Argument In	Array of signal pulse centers ranges in distance units. For NUMPRO signals on the signal bus.
SGPW	Argument In	Array of signal pulse widths in time. For NUMPRO signals on the signal bus.
SGSV	Argument In	Array of sum-channel signal complex voltages, NUMPRO in number. As called by WFTCPI, this is the sum-channel after doppler filtering (SGSVFL in WFTCPI).
TRGC	Return from PRTION	Center of range gate on time base.
TRGEP(-)	Return from PRTION	Array of locations (in time) of discrete pulse edges found in range gate.
TRGW	Argument In	Range gate width in time.
VCNTLA	Argument In	Control voltage for the special detailed receiver using log amplifiers.
VLTOUT	Return from ARNGTB	Negative of refined range error returned by special detailed receiver (see Part II).
WTERLY	Argument In	Weight to be given the integrated voltage signal in the early gate in computing the range error.
WTLATE	Argument In	Weight to be given the integrated voltage signal in the late gate in computing the range error.

TABLE 2.28-6b. Subroutine RNGDSC—Output Data.

Name	Source	Description
AGCLVL	Argument to ARNGTB	AGC level input for range error refinement by special detailed receiver (see Part II).
BPL	Argument Out; Argument to PRTION, INTGRT	The left (early) edge of the blanking pulse, if the blanking pulse is on, trimmed to the left edge of the range gate if necessary.
BPR	Argument Out; Argument to PRTION, INTGRT	The right (late) edge of the blanking pulse, if the blanking pulse is on, trimmed to the right edge of the range gate if necessary.
DELTAT	Argument to ARNGTB	Time equivalent of range error (before refinement by special detailed receiver (see Part II)).
IBLANK	Argument Out; Argument to PRTION, INTGRT	Flag indicating whether the blanking pulse is on (=1, on; =0, off), turned off if the pulse is entirely out of the gate.
LOGAMP	Argument to INTGRT	Flag indication whether the special detailed receiver using log amplifiers is in use (=2, means the special receiver is in use).
NEP	Argument to INTGRT	Number of discrete pulse edges found in range gate.
NUMPRO	Argument to PRTION, INTGRT	Number of signals on signal bus.
PTAR(-,-)	Argument to PRTION, INTGRT	Array of pulse time of arrival for center and left and right edges of all pulses on signal bus.

TABLE 2.28-6b. Subroutine RNGDSC—Output Data. (Contd.)

Name	Source	Description
RERROR	Argument Out	Range error in distance units.
RGATE	Argument to PRTION	Current setting of current radar range-gate center.
SGPW(-)	Argument to PRTION	Array of signal pulse widths on signal bus.
SGSV(-)	Argument to INTGRT	Array of sum-channel signal complex voltages, NUMPRO in number. As called by WFTCPI, this is the sum-channel after doppler filtering (SGSVFL in WFTCPI).
TRGC	Argument to INTGRT	Center of range gate on time base.
TRGEP(-)	Argument to INTGRT	Array of locations (in time) of discrete pulse edges found in range gate.
TRGW(-)	Argument to PRTION	Array of radar range-gate widths in time.
VCNTLA	Argument to INTGRT	Control voltage for the special detailed receiver using log amplifiers.

TABLE 2.28-7a. Subroutine PRTION—Input Data.

Name	Source	Description
BPL	Argument In	On input, the left (early) edge of the blanking pulse, if the blanking pulse is on.
BPR	Argument In	On input, the right (late) edge of the blanking pulse, if the blanking pulse is on.
IBLANK	Argument In	Flag indicating whether the blanking pulse is on (=1, on; =0, off).
NUMPRO	Argument In	Number of signals on signal bus.
PTAR	Argument In	On input, (ISIG,3) element contains center (in time) of ISIG-th signal pulse (ISIG=1,NUMPRO).
RGATE	Argument In	Location of center of range gate in distance.
SGPW	Argument In	Array of signal pulse widths, NUMPRO in number.
TRGW	Argument In	Width of range gate in time.

TABLE 2.28-7b. Subroutine PRTION—Output Data.

Name	Source	Description
BPL	Argument Out	On output, the left (early) edge of the blanking pulse, if the blanking pulse is on, may be trimmed to the left edge of the range gate if the pulse extends farther left.
BPR	Argument Out	On output, the right (late) edge of the blanking pulse, if the blanking pulse is on, may be trimmed to the right edge of the range gate if the pulse extends farther right.
IBLANK	Argument Out	Flag indicating whether the blanking pulse is on (=1, on; =0, off). If =1 on input, will be turned off if none of blanking pulse falls inside the range gate.
NEP	Argument Out	Number of signal pulse edges found in the range gate, with the left and right edges of the gate itself counting as two edges in NEP.

TABLE 2.28-7b. Subroutine PRTION—Output Data. (Contd.)

Name	Source	Description
PTAR	Argument Out	On output, (ISIG,1) element contains early (left) edge and (ISIG,2) element contains late (right) edge (in time) of ISIG-th signal pulse (ISIG=1,NUMPRO).
TRGC	Argument Out	Location of the range gate center in time.
TRGEP	Argument Out	Array of ordered signal pulse edges (in time) found in partitioning the range gate; number is NEP, with number 1 being the left (early) edge of range gate and NEP being the right (late) edge of the gate.

TABLE 2.28-8a. Subroutine HEAPIN—Input Data.

Name	Source	Description
ARRAY(-)	Argument In	Array holding the heap accumulated to date.
ELENEW	Argument In	New element to be added to heap (list).
NUMELE	Argument In	Number of elements in heap on entry to date.

TABLE 2.28-8b. Subroutine HEAPIN—Output Data.

Name	Source	Description
ARRAY(-)	Argument Out	Array holding the heap accumulated so far, updated for the element just added.
NUMELE	Argument Out	Number of elements in heap on return from HEAPIN, updated for the element just added.

TABLE 2.28-9a. Subroutine HEAPST—Input Data.

Name	Source	Description
ARRAY(-)	Argument In	Array holding the list accumulated.
NUMELE	Argument In	Number of elements in accumulated in array.

TABLE 2.28-9b. Subroutine HEAPST—Output Data.

Name	Source	Description
ARRAY(-)	Argument Out	Array holding the list, reconstructed in the form of a heap and sorted in increasing order of value.

TABLE 2.28-10a. Subroutine HEAPSR—Input Data.

Name	Source	Description
ARRAY(-)	Argument In	Array holding the list accumulated.
NUMELE	Argument In	Number of elements in accumulated in array.

TABLE 2.28-10b. Subroutine HEAPSR—Output Data.

Name	Source	Description
ARRAY(-)	Argument Out	Array holding the list, reconstructed in the form of a heap.

TABLE 2.28-11a. Subroutine INTGRT—Input Data.

Name	Source	Description
BPL	Argument In	The left (early) edge of the blanking pulse, if the blanking pulse is on, trimmed to the left edge of the range gate if necessary.
BPR	Argument In	The right (late) edge of the blanking pulse, if the blanking pulse is on, trimmed to the right edge of the range gate if necessary.
IBLANK	Argument In	Flag indicating whether the blanking pulse is on (=1, on; =0, off), turned off if the pulse is entirely out of the gate.
LOGAMP	Argument In	Flag indication whether the special detailed receiver using log amplifiers is in use (=2, means the special receiver is in use).
NEP	Argument In	Number of signal pulse edges found in the range gate, with the left and right edges of the gate itself counting as two edges in NEP.
NUMPRO	Argument In	Number of signals on signal bus.
PTAR	Argument In	Array of signal pulse centers and left and right edges. For the ISIG-th signal pulse (ISIG=1,NUMPRO), (ISIG,3) element contains the center (in time), the (ISIG,1) element contains the early (left) edge, and the (ISIG,2) element contains the late (right) edge.
RGATE	Argument In	Location of center of range gate in distance.
RVLREC	Argument In	Array of sum-channel signal complex voltages, NUMPRO in number. As called by RNGDSC, this is the sum-channel after doppler filtering.
TRGC	Argument In	Location of the range gate center in time.
TRGEP	Argument In	Array of ordered signal pulse edges (in time) found in partitioning the range gate; number is NEP, with number 1 being the left (early) edge of range gate and NEP being the right (late) edge of the gate.
TRGW	Argument In	Width of range gate in time.
VCNTLA	Argument In	Control voltage for the special detailed receiver using log amplifiers.

TABLE 2.28-11b. Subroutine INTGRT—Output Data.

Name	Source	Description
EARLY	Argument Out	Magnitude of the integrated total signal voltage in the early gate of the split range gate.
LATE	Argument Out	Magnitude of the integrated total signal voltage in the late gate of the split range gate.

TABLE 2.28-12a. Subroutine UPD8RG—Input Data.

Table 2.28-12a.		
Name	**Source**	**Description**
GRPDUR	Argument In	Duration time of current group in waveform.
IRADFL	Common FLAGS	Current radar type in use.
IRSTYP(-)	Common SVOVAR	Array of range servo type indices.
ITRACK(-)	Common FLAGS	Array of target tracking flags, indicating whether radar is tracking currently.
RGATED(-)	Common GRADAR; Return from RGHKF, or RGHF	Array of current range-gate rates for all radar types. On return from indicated subroutines, is the updated range-rate corrected for current range error.
RGATES(-)	Common GRADAR; Return from SVORNG, RNGSVA, RGHKF, SVORSA, RGHF, or RNGSV1	Array of current range-gate settings for all radar types. On return from indicated subroutines, is the updated range corrected for current range error.
TIMEG	Common GRADAR	Current simulation time.
TOA	Common SUMARY	Time of target acquisition for tracking.
WRNGER	Argument In	Range tracking error from receiver.

TABLE 2.28-12b. Subroutine UPD8RG—Output Data.

Name	**Source**	**Description**
GRPDUR	Argument to RGHKF, or RGHF	Duration time of current group in waveform.
ITRACK(-)	Argument to RNGSVA, RGHKF, SVORSA, RGHF	Array of target tracking flags, indicating whether radar is tracking currently.
RGATED(-)	Common GRADAR; Argument to RGHKF, or RGHF	Array of current range-gate rates for all radar types, updated after return from indicated routines to correct for current range error. (Update of range rates done via this path only for ghk and gh track filters; the servo routines update RGATED directly in GRADAR.)
RGATES(-)	Common GRADAR; Argument to SVORNG, RNGSVA, RGHKF, SVORSA, RGHF, or RNGSV1	Array of current range-gate settings for all radar types, updated after return from indicated routines to correct for current range error.
TIMEG	Argument to RGHKF	Current simulation time.
TIMET	Common GRADAR; Argument to RNGSVA	Current time in tracking mode.
WRNGER	Argument to SVORNG, RNGSVA, RGHKF, SVORSA, RGHF, or RNGSV1	Range tracking error from receiver.

TABLE 2.28-13a. Subroutine RGHKF—Input Data.

Name	Source	Description
ASLR	Common RGHK; Return from GHK	Previous value of smoothed range acceleration, at beginning of current group. On return from GHK, value is updated for end of current group.
ASR	Common RGHK; Return from GHK	Smoothed range acceleration, at beginning of current group. On return from GHK, value is updated for end of current group.
DXALR	Common RGHK; Return from GHK	Range increment due to acceleration, at beginning of current group. On return from GHK, value is updated for end of current group.
DXVLR	Common RGHK; Return from GHK	Range increment due to velocity, at beginning of current group. On return from GHK, value is updated for end of current group.
EXR	Common RGHK; Return from GHK	Range estimate error, at beginning of current group. On return from GHK, value is updated for end of current group.
GHKGR(-)	Common RGHK	Array of early and late gains g for range ghk track filter.
GRPDUR	Argument In	Duration time of current group in waveform.
ILOCK	Argument In	Target tracking flag, indicating whether radar is tracking currently.
RGATED	Argument In	Current range-gate rate for current radar type.
RGATES	Argument In	Current range-gate setting for current radar type.
RNGERR	Argument In	Range tracking error from receiver.
TIME	Argument In	Current simulation time.
TSGHKR	Common RGHK	Time for switchover between early and late gains g for range ghk track filter.
VSALR	Common RGHK; Return from GHK	Range rate increment due to acceleration, at beginning of current group. On return from GHK, value is updated for end of current group.
VSLR	Common RGHK; Return from GHK	Previous value of smoothed range rate, at beginning of current group. On return from GHK, value is updated for end of current group.
VSR	Common RGHK; Return from GHK	Smoothed range rate, at beginning of current group. On return from GHK, value is updated for end of current group.
XPR	Common RGHK; Return from GHK	Range prediction, at beginning of current group. On return from GHK, value is updated for end of current group.
XSLR	Common RGHK; Return from GHK	Previous value of smoothed range, at beginning of current group. On return from GHK, value is updated for end of current group.
XSR	Common RGHK; Return from GHK	Smoothed range, at beginning of current group. On return from GHK, value is updated for end of current group.

TABLE 2.28-13b. Subroutine RGHKF—Output Data.

Name	Source	Description
ASLR	Argument to GHK; Common RGHK	Previous value of smoothed range acceleration, at beginning of current group. On return from GHK, value is updated for end of current group.
ASR	Argument to GHK; Common RGHK	Smoothed range acceleration, at beginning of current group. On return from GHK, value is updated for end of current group.

TABLE 2.28-13b. Subroutine RGHKF—Output Data. (Contd.)

Name	Source	Description
DXALR	Argument to GHK; Common RGHK	Range increment due to acceleration, at beginning of current group. On return from GHK, value is updated for end of current group.
DXVLR	Argument to GHK; Common RGHK	Range increment due to velocity, at beginning of current group. On return from GHK, value is updated for end of current group.
EXR	Argument to GHK; Common RGHK	Range estimate error, at beginning of current group. On return from GHK, value is updated for end of current group.
GHKGR(-)	Argument to GHK	Array of early and late gains g for range ghk track filter.
RGATED	Argument Out	Current range-gate rate for current radar type, updated to correct for current range error.
RGATES(-)	Argument Out	Current range-gate setting for current radar type, updated to correct for current range error.
TSGHKR	Argument to GHK	Time for switchover between early and late gains g for range ghk track filter.
VSALR	Argument to GHK; Common RGHK	Range rate increment due to acceleration, at beginning of current group. On return from GHK, value is updated for end of current group.
VSLR	Argument to GHK; Common RGHK	Previous value of smoothed range rate, at beginning of current group. On return from GHK, value is updated for end of current group.
VSR	Argument to GHK; Common RGHK	Smoothed range rate, at beginning of current group. On return from GHK, value is updated for end of current group.
XMR	Argument to GHK; Common RGHK	Measured range for current group.
XPR	Argument to GHK; Common RGHK	Range prediction, at beginning of current group. On return from GHK, value is updated for end of current group.
XSLR	Argument to GHK; Common RGHK	Previous value of smoothed range, at beginning of current group. On return from GHK, value is updated for end of current group.
XSR	Argument to GHK; Common RGHK	Smoothed range, at beginning of current group. On return from GHK, value is updated for end of current group.

TABLE 2.28-14a. Subroutine GHK—Input Data.

Name	Source	Description
ASL	Argument In	Smoothed coordinate acceleration at previous sample.
DTGHK	Argument In	Sample time interval.
DXAL	Argument In	Coordinate increment due to smoothed acceleration.
DXVL	Argument In	Coordinate increment due to smoothed velocity.
GHKG(-)	Argument In	Two-element array of early and late coordinate gains for g-h-k track filter.
TIME	Argument In	Time in tracking phase.
TSWGHK	Argument In	Time for switch-over between early and late gains for g-h-k track filter.
VSAL	Argument In	Coordinate velocity increment due to smoothed acceleration.
VSL	Argument In	Smoothed coordinate velocity at previous sample.
XM	Argument In	Measured coordinate.
XSL	Argument In	Smoothed coordinate at previous sample.

TABLE 2.28-14b. Subroutine GHK—Output Data.

Name	Source	Description
AS	Argument Out	Smoothed coordinate acceleration.
ASL	Argument Out	Smoothed coordinate acceleration at previous sample.
DXAL	Argument Out	Coordinate increment due to smoothed acceleration.
DXVL	Argument Out	Coordinate increment due to smoothed velocity.
EX	Argument Out	Coordinate estimate error.
VS	Argument Out	Smoothed coordinate velocity.
VSAL	Argument Out	Coordinate velocity increment due to smoothed acceleration.
VSL	Argument Out	Smoothed coordinate velocity at previous sample.
XM	Argument Out	Measured coordinate.
XP	Argument Out	Predicted coordinate.
XS	Argument Out	Smoothed coordinate.
XSL	Argument Out	Smoothed coordinate at previous sample.

2.28.4 Assumptions and Limitations

In the range error measurement subelement, the range-gate and all signals are represented as rectangular pulses, with range-track related characteristics of center location, width, magnitude, and phase. This limits the capability to handle pulse-shaping and taking into account any other non-rectangular features of gates and signals. In the range-track F.E., the capability to weight the early and late gates differently compensates somewhat for this limitation.

Range-track related ECCM has been enhanced in ESAMS 2.7 over the previous “street” release 2.6.2. This capability, along with the design approach and software design for the detailed two channel monopulse receiver, is discussed in the classified addendum.

For the range-correction subelement, ESAMS 2.7 now has three additional servo models beyond the generic improved type I/II servos, including the servo RNGSV1 coming out of the SMART project (see classified addendum). Similarly, there are two track filter models–the ghk and its more restricted sibling the gh filter–which offer considerable versatility in tailoring ESAMS to a particular system. The ghk and gh filter models are limited to just one independent parameter, g, with the other filter parameters (h and k) being derived from by fixed relations. However, ESAMS 2.7 does permit g to have two different values depending on how long the tracker has been tracking a target; this permits a less demanding filter for the initial tracking, with a more demanding filter after the track is well established.

At this time, this FE only addresses the requirements for the ghk filter. The requirements, design approach, and software design for the additional filters will be included in future updates.

No other limitations are known.